

**NVIDIA Performance Primitives (NPP)**  
Version 5.0

September 7, 2012



# Contents

<b>1</b>	<b>NVIDIA Performance Primitives</b>	<b>1</b>
1.1	What is NPP? . . . . .	1
1.2	Documentation . . . . .	1
1.3	Technical Specifications . . . . .	2
1.4	Files . . . . .	2
1.4.1	Header Files . . . . .	2
1.4.2	Library Files . . . . .	2
1.5	Supported NVIDIA Hardware . . . . .	3
<b>2</b>	<b>General API Conventions</b>	<b>5</b>
2.1	Memory Management . . . . .	6
2.1.1	Scratch Buffer and Host Pointer . . . . .	6
2.2	Function Naming . . . . .	6
2.3	Integer Result Scaling . . . . .	7
<b>3</b>	<b>Signal-Processing Specific API Conventions</b>	<b>9</b>
3.1	Signal Data . . . . .	10
3.1.1	Parameter Names for Signal Data . . . . .	10
3.1.1.1	Source Signal Pointer . . . . .	10
3.1.1.2	Destination Signal Pointer . . . . .	10
3.1.1.3	In-Place Signal Pointer . . . . .	10
3.1.2	Signal Data Alignment Requirements . . . . .	11
3.1.3	Signal Data Related Error Codes . . . . .	11
3.2	Signal Length . . . . .	11
3.2.1	Length Related Error Codes . . . . .	11
<b>4</b>	<b>Imaging-Processing Specific API Conventions</b>	<b>13</b>
4.1	Function Naming . . . . .	14
4.2	Image Data . . . . .	14

4.2.1	Line Step	15
4.2.2	Parameter Names for Image Data	15
4.2.2.1	Passing Source-Image Data	15
4.2.2.2	Passing Destination-Image Data	16
4.2.2.3	Passing In-Place Image Data	18
4.2.2.4	Passing Mask-Image Data	18
4.2.2.5	Passing Channel-of-Interest Data	18
4.2.3	Image Data Alignment Requirements	18
4.2.4	Image Data Related Error Codes	19
4.3	Region-of-Interest (ROI)	19
4.3.1	ROI Related Error Codes	20
4.4	Masked Operation	20
4.5	Channel-of-Interest API	20
4.5.1	Select-Channel Source-Image Pointer	20
4.5.2	Select-Channel Source-Image	20
4.5.3	Select-Channel Destination-Image Pointer	21
<b>5</b>	<b>Module Index</b>	<b>23</b>
5.1	Modules	23
<b>6</b>	<b>Data Structure Index</b>	<b>27</b>
6.1	Data Structures	27
<b>7</b>	<b>Module Documentation</b>	<b>29</b>
7.1	NPP Core	29
7.1.1	Detailed Description	29
7.1.2	Function Documentation	30
7.1.2.1	nppGetGpuComputeCapability	30
7.1.2.2	nppGetGpuName	30
7.1.2.3	nppGetGpuNumSMs	30
7.1.2.4	nppGetLibVersion	30
7.1.2.5	nppGetMaxThreadsPerBlock	30
7.1.2.6	nppGetMaxThreadsPerSM	31
7.1.2.7	nppGetStream	31
7.1.2.8	nppSetStream	31
7.2	NPP Type Definitions and Constants	32
7.2.1	Define Documentation	36
7.2.1.1	NPP_MAX_16S	36

7.2.1.2	NPP_MAX_16U	36
7.2.1.3	NPP_MAX_32S	36
7.2.1.4	NPP_MAX_32U	36
7.2.1.5	NPP_MAX_64S	36
7.2.1.6	NPP_MAX_64U	36
7.2.1.7	NPP_MAX_8S	36
7.2.1.8	NPP_MAX_8U	36
7.2.1.9	NPP_MAXABS_32F	37
7.2.1.10	NPP_MAXABS_64F	37
7.2.1.11	NPP_MIN_16S	37
7.2.1.12	NPP_MIN_16U	37
7.2.1.13	NPP_MIN_32S	37
7.2.1.14	NPP_MIN_32U	37
7.2.1.15	NPP_MIN_64S	37
7.2.1.16	NPP_MIN_64U	37
7.2.1.17	NPP_MIN_8S	37
7.2.1.18	NPP_MIN_8U	37
7.2.1.19	NPP_MINABS_32F	37
7.2.1.20	NPP_MINABS_64F	38
7.2.2	Enumeration Type Documentation	38
7.2.2.1	NppCmpOp	38
7.2.2.2	NppGpuComputeCapability	38
7.2.2.3	NppiAlphaOp	38
7.2.2.4	NppiAxis	39
7.2.2.5	NppiBorderType	39
7.2.2.6	NppiInterpolationMode	39
7.2.2.7	NppRoundMode	39
7.2.2.8	NppStatus	40
7.2.2.9	NppsZCType	41
7.3	Basic NPP Data Types	42
7.3.1	Typedef Documentation	43
7.3.1.1	Npp16s	43
7.3.1.2	Npp16u	43
7.3.1.3	Npp32f	43
7.3.1.4	Npp32s	43
7.3.1.5	Npp32u	43

7.3.1.6	Npp64f	43
7.3.1.7	Npp64s	43
7.3.1.8	Npp64u	43
7.3.1.9	Npp8s	44
7.3.1.10	Npp8u	44
7.4	NPP Image Processing	45
7.5	Arithmetic and Logical Operations	46
7.6	Arithmetic Operations	47
7.7	AddC	49
7.7.1	Detailed Description	54
7.7.2	Function Documentation	54
7.7.2.1	nppiAddC_16s_AC4IRSfs	54
7.7.2.2	nppiAddC_16s_AC4RSfs	54
7.7.2.3	nppiAddC_16s_C1IRSfs	54
7.7.2.4	nppiAddC_16s_C1RSfs	55
7.7.2.5	nppiAddC_16s_C3IRSfs	55
7.7.2.6	nppiAddC_16s_C3RSfs	56
7.7.2.7	nppiAddC_16s_C4IRSfs	56
7.7.2.8	nppiAddC_16s_C4RSfs	56
7.7.2.9	nppiAddC_16sc_AC4IRSfs	57
7.7.2.10	nppiAddC_16sc_AC4RSfs	57
7.7.2.11	nppiAddC_16sc_C1IRSfs	58
7.7.2.12	nppiAddC_16sc_C1RSfs	58
7.7.2.13	nppiAddC_16sc_C3IRSfs	58
7.7.2.14	nppiAddC_16sc_C3RSfs	59
7.7.2.15	nppiAddC_16u_AC4IRSfs	59
7.7.2.16	nppiAddC_16u_AC4RSfs	60
7.7.2.17	nppiAddC_16u_C1IRSfs	60
7.7.2.18	nppiAddC_16u_C1RSfs	60
7.7.2.19	nppiAddC_16u_C3IRSfs	61
7.7.2.20	nppiAddC_16u_C3RSfs	61
7.7.2.21	nppiAddC_16u_C4IRSfs	62
7.7.2.22	nppiAddC_16u_C4RSfs	62
7.7.2.23	nppiAddC_32f_AC4IR	62
7.7.2.24	nppiAddC_32f_AC4R	63
7.7.2.25	nppiAddC_32f_C1IR	63

7.7.2.26	<code>nppiAddC_32f_C1R</code>	63
7.7.2.27	<code>nppiAddC_32f_C3IR</code>	64
7.7.2.28	<code>nppiAddC_32f_C3R</code>	64
7.7.2.29	<code>nppiAddC_32f_C4IR</code>	64
7.7.2.30	<code>nppiAddC_32f_C4R</code>	65
7.7.2.31	<code>nppiAddC_32fc_AC4IR</code>	65
7.7.2.32	<code>nppiAddC_32fc_AC4R</code>	65
7.7.2.33	<code>nppiAddC_32fc_C1IR</code>	66
7.7.2.34	<code>nppiAddC_32fc_C1R</code>	66
7.7.2.35	<code>nppiAddC_32fc_C3IR</code>	66
7.7.2.36	<code>nppiAddC_32fc_C3R</code>	67
7.7.2.37	<code>nppiAddC_32fc_C4IR</code>	67
7.7.2.38	<code>nppiAddC_32fc_C4R</code>	67
7.7.2.39	<code>nppiAddC_32s_C1IRSfs</code>	68
7.7.2.40	<code>nppiAddC_32s_C1RSfs</code>	68
7.7.2.41	<code>nppiAddC_32s_C3IRSfs</code>	68
7.7.2.42	<code>nppiAddC_32s_C3RSfs</code>	69
7.7.2.43	<code>nppiAddC_32sc_AC4IRSfs</code>	69
7.7.2.44	<code>nppiAddC_32sc_AC4RSfs</code>	70
7.7.2.45	<code>nppiAddC_32sc_C1IRSfs</code>	70
7.7.2.46	<code>nppiAddC_32sc_C1RSfs</code>	70
7.7.2.47	<code>nppiAddC_32sc_C3IRSfs</code>	71
7.7.2.48	<code>nppiAddC_32sc_C3RSfs</code>	71
7.7.2.49	<code>nppiAddC_8u_AC4IRSfs</code>	72
7.7.2.50	<code>nppiAddC_8u_AC4RSfs</code>	72
7.7.2.51	<code>nppiAddC_8u_C1IRSfs</code>	72
7.7.2.52	<code>nppiAddC_8u_C1RSfs</code>	73
7.7.2.53	<code>nppiAddC_8u_C3IRSfs</code>	73
7.7.2.54	<code>nppiAddC_8u_C3RSfs</code>	73
7.7.2.55	<code>nppiAddC_8u_C4IRSfs</code>	74
7.7.2.56	<code>nppiAddC_8u_C4RSfs</code>	74
7.8	MulC	75
7.8.1	Detailed Description	80
7.8.2	Function Documentation	80
7.8.2.1	<code>nppiMulC_16s_AC4IRSfs</code>	80
7.8.2.2	<code>nppiMulC_16s_AC4RSfs</code>	80

7.8.2.3	nppiMulC_16s_C1IRSfs	81
7.8.2.4	nppiMulC_16s_C1RSfs	81
7.8.2.5	nppiMulC_16s_C3IRSfs	81
7.8.2.6	nppiMulC_16s_C3RSfs	82
7.8.2.7	nppiMulC_16s_C4IRSfs	82
7.8.2.8	nppiMulC_16s_C4RSfs	82
7.8.2.9	nppiMulC_16sc_AC4IRSfs	83
7.8.2.10	nppiMulC_16sc_AC4RSfs	83
7.8.2.11	nppiMulC_16sc_C1IRSfs	84
7.8.2.12	nppiMulC_16sc_C1RSfs	84
7.8.2.13	nppiMulC_16sc_C3IRSfs	84
7.8.2.14	nppiMulC_16sc_C3RSfs	85
7.8.2.15	nppiMulC_16u_AC4IRSfs	85
7.8.2.16	nppiMulC_16u_AC4RSfs	86
7.8.2.17	nppiMulC_16u_C1IRSfs	86
7.8.2.18	nppiMulC_16u_C1RSfs	86
7.8.2.19	nppiMulC_16u_C3IRSfs	87
7.8.2.20	nppiMulC_16u_C3RSfs	87
7.8.2.21	nppiMulC_16u_C4IRSfs	88
7.8.2.22	nppiMulC_16u_C4RSfs	88
7.8.2.23	nppiMulC_32f_AC4IR	88
7.8.2.24	nppiMulC_32f_AC4R	89
7.8.2.25	nppiMulC_32f_C1IR	89
7.8.2.26	nppiMulC_32f_C1R	89
7.8.2.27	nppiMulC_32f_C3IR	90
7.8.2.28	nppiMulC_32f_C3R	90
7.8.2.29	nppiMulC_32f_C4IR	90
7.8.2.30	nppiMulC_32f_C4R	91
7.8.2.31	nppiMulC_32fc_AC4IR	91
7.8.2.32	nppiMulC_32fc_AC4R	91
7.8.2.33	nppiMulC_32fc_C1IR	92
7.8.2.34	nppiMulC_32fc_C1R	92
7.8.2.35	nppiMulC_32fc_C3IR	92
7.8.2.36	nppiMulC_32fc_C3R	93
7.8.2.37	nppiMulC_32fc_C4IR	93
7.8.2.38	nppiMulC_32fc_C4R	93

7.8.2.39	nppiMulC_32s_C1IRSfs	94
7.8.2.40	nppiMulC_32s_C1RSfs	94
7.8.2.41	nppiMulC_32s_C3IRSfs	94
7.8.2.42	nppiMulC_32s_C3RSfs	95
7.8.2.43	nppiMulC_32sc_AC4IRSfs	95
7.8.2.44	nppiMulC_32sc_AC4RSfs	96
7.8.2.45	nppiMulC_32sc_C1IRSfs	96
7.8.2.46	nppiMulC_32sc_C1RSfs	96
7.8.2.47	nppiMulC_32sc_C3IRSfs	97
7.8.2.48	nppiMulC_32sc_C3RSfs	97
7.8.2.49	nppiMulC_8u_AC4IRSfs	98
7.8.2.50	nppiMulC_8u_AC4RSfs	98
7.8.2.51	nppiMulC_8u_C1IRSfs	98
7.8.2.52	nppiMulC_8u_C1RSfs	99
7.8.2.53	nppiMulC_8u_C3IRSfs	99
7.8.2.54	nppiMulC_8u_C3RSfs	99
7.8.2.55	nppiMulC_8u_C4IRSfs	100
7.8.2.56	nppiMulC_8u_C4RSfs	100
7.9	MulCScale	101
7.9.1	Detailed Description	102
7.9.2	Function Documentation	102
7.9.2.1	nppiMulCScale_16u_AC4IR	102
7.9.2.2	nppiMulCScale_16u_AC4R	103
7.9.2.3	nppiMulCScale_16u_C1IR	103
7.9.2.4	nppiMulCScale_16u_C1R	103
7.9.2.5	nppiMulCScale_16u_C3IR	104
7.9.2.6	nppiMulCScale_16u_C3R	104
7.9.2.7	nppiMulCScale_16u_C4IR	104
7.9.2.8	nppiMulCScale_16u_C4R	105
7.9.2.9	nppiMulCScale_8u_AC4IR	105
7.9.2.10	nppiMulCScale_8u_AC4R	105
7.9.2.11	nppiMulCScale_8u_C1IR	106
7.9.2.12	nppiMulCScale_8u_C1R	106
7.9.2.13	nppiMulCScale_8u_C3IR	106
7.9.2.14	nppiMulCScale_8u_C3R	107
7.9.2.15	nppiMulCScale_8u_C4IR	107

7.9.2.16	nppiMulCScale_8u_C4R	107
7.10	SubC	108
7.10.1	Detailed Description	113
7.10.2	Function Documentation	113
7.10.2.1	nppiSubC_16s_AC4IRSfs	113
7.10.2.2	nppiSubC_16s_AC4RSfs	113
7.10.2.3	nppiSubC_16s_C1IRSfs	113
7.10.2.4	nppiSubC_16s_C1RSfs	114
7.10.2.5	nppiSubC_16s_C3IRSfs	114
7.10.2.6	nppiSubC_16s_C3RSfs	115
7.10.2.7	nppiSubC_16s_C4IRSfs	115
7.10.2.8	nppiSubC_16s_C4RSfs	115
7.10.2.9	nppiSubC_16sc_AC4IRSfs	116
7.10.2.10	nppiSubC_16sc_AC4RSfs	116
7.10.2.11	nppiSubC_16sc_C1IRSfs	117
7.10.2.12	nppiSubC_16sc_C1RSfs	117
7.10.2.13	nppiSubC_16sc_C3IRSfs	117
7.10.2.14	nppiSubC_16sc_C3RSfs	118
7.10.2.15	nppiSubC_16u_AC4IRSfs	118
7.10.2.16	nppiSubC_16u_AC4RSfs	119
7.10.2.17	nppiSubC_16u_C1IRSfs	119
7.10.2.18	nppiSubC_16u_C1RSfs	119
7.10.2.19	nppiSubC_16u_C3IRSfs	120
7.10.2.20	nppiSubC_16u_C3RSfs	120
7.10.2.21	nppiSubC_16u_C4IRSfs	121
7.10.2.22	nppiSubC_16u_C4RSfs	121
7.10.2.23	nppiSubC_32f_AC4IR	121
7.10.2.24	nppiSubC_32f_AC4R	122
7.10.2.25	nppiSubC_32f_C1IR	122
7.10.2.26	nppiSubC_32f_C1R	122
7.10.2.27	nppiSubC_32f_C3IR	123
7.10.2.28	nppiSubC_32f_C3R	123
7.10.2.29	nppiSubC_32f_C4IR	123
7.10.2.30	nppiSubC_32f_C4R	124
7.10.2.31	nppiSubC_32fc_AC4IR	124
7.10.2.32	nppiSubC_32fc_AC4R	124

---

7.10.2.33	nppiSubC_32fc_C1IR	125
7.10.2.34	nppiSubC_32fc_C1R	125
7.10.2.35	nppiSubC_32fc_C3IR	125
7.10.2.36	nppiSubC_32fc_C3R	126
7.10.2.37	nppiSubC_32fc_C4IR	126
7.10.2.38	nppiSubC_32fc_C4R	126
7.10.2.39	nppiSubC_32s_C1IRSfs	127
7.10.2.40	nppiSubC_32s_C1RSfs	127
7.10.2.41	nppiSubC_32s_C3IRSfs	127
7.10.2.42	nppiSubC_32s_C3RSfs	128
7.10.2.43	nppiSubC_32sc_AC4IRSfs	128
7.10.2.44	nppiSubC_32sc_AC4RSfs	129
7.10.2.45	nppiSubC_32sc_C1IRSfs	129
7.10.2.46	nppiSubC_32sc_C1RSfs	129
7.10.2.47	nppiSubC_32sc_C3IRSfs	130
7.10.2.48	nppiSubC_32sc_C3RSfs	130
7.10.2.49	nppiSubC_8u_AC4IRSfs	131
7.10.2.50	nppiSubC_8u_AC4RSfs	131
7.10.2.51	nppiSubC_8u_C1IRSfs	131
7.10.2.52	nppiSubC_8u_C1RSfs	132
7.10.2.53	nppiSubC_8u_C3IRSfs	132
7.10.2.54	nppiSubC_8u_C3RSfs	132
7.10.2.55	nppiSubC_8u_C4IRSfs	133
7.10.2.56	nppiSubC_8u_C4RSfs	133
7.11	DivC	134
7.11.1	Detailed Description	139
7.11.2	Function Documentation	139
7.11.2.1	nppiDivC_16s_AC4IRSfs	139
7.11.2.2	nppiDivC_16s_AC4RSfs	139
7.11.2.3	nppiDivC_16s_C1IRSfs	140
7.11.2.4	nppiDivC_16s_C1RSfs	140
7.11.2.5	nppiDivC_16s_C3IRSfs	140
7.11.2.6	nppiDivC_16s_C3RSfs	141
7.11.2.7	nppiDivC_16s_C4IRSfs	141
7.11.2.8	nppiDivC_16s_C4RSfs	141
7.11.2.9	nppiDivC_16sc_AC4IRSfs	142

---

7.11.2.10 nppiDivC_16sc_AC4RSfs . . . . .	142
7.11.2.11 nppiDivC_16sc_C1IRSfs . . . . .	143
7.11.2.12 nppiDivC_16sc_C1RSfs . . . . .	143
7.11.2.13 nppiDivC_16sc_C3IRSfs . . . . .	143
7.11.2.14 nppiDivC_16sc_C3RSfs . . . . .	144
7.11.2.15 nppiDivC_16u_AC4IRSfs . . . . .	144
7.11.2.16 nppiDivC_16u_AC4RSfs . . . . .	145
7.11.2.17 nppiDivC_16u_C1IRSfs . . . . .	145
7.11.2.18 nppiDivC_16u_C1RSfs . . . . .	145
7.11.2.19 nppiDivC_16u_C3IRSfs . . . . .	146
7.11.2.20 nppiDivC_16u_C3RSfs . . . . .	146
7.11.2.21 nppiDivC_16u_C4IRSfs . . . . .	147
7.11.2.22 nppiDivC_16u_C4RSfs . . . . .	147
7.11.2.23 nppiDivC_32f_AC4IR . . . . .	147
7.11.2.24 nppiDivC_32f_AC4R . . . . .	148
7.11.2.25 nppiDivC_32f_C1IR . . . . .	148
7.11.2.26 nppiDivC_32f_C1R . . . . .	148
7.11.2.27 nppiDivC_32f_C3IR . . . . .	149
7.11.2.28 nppiDivC_32f_C3R . . . . .	149
7.11.2.29 nppiDivC_32f_C4IR . . . . .	149
7.11.2.30 nppiDivC_32f_C4R . . . . .	150
7.11.2.31 nppiDivC_32fc_AC4IR . . . . .	150
7.11.2.32 nppiDivC_32fc_AC4R . . . . .	150
7.11.2.33 nppiDivC_32fc_C1IR . . . . .	151
7.11.2.34 nppiDivC_32fc_C1R . . . . .	151
7.11.2.35 nppiDivC_32fc_C3IR . . . . .	151
7.11.2.36 nppiDivC_32fc_C3R . . . . .	152
7.11.2.37 nppiDivC_32fc_C4IR . . . . .	152
7.11.2.38 nppiDivC_32fc_C4R . . . . .	152
7.11.2.39 nppiDivC_32s_C1IRSfs . . . . .	153
7.11.2.40 nppiDivC_32s_C1RSfs . . . . .	153
7.11.2.41 nppiDivC_32s_C3IRSfs . . . . .	153
7.11.2.42 nppiDivC_32s_C3RSfs . . . . .	154
7.11.2.43 nppiDivC_32sc_AC4IRSfs . . . . .	154
7.11.2.44 nppiDivC_32sc_AC4RSfs . . . . .	155
7.11.2.45 nppiDivC_32sc_C1IRSfs . . . . .	155

---

7.11.2.46	nppiDivC_32sc_C1RSfs	155
7.11.2.47	nppiDivC_32sc_C3IRSfs	156
7.11.2.48	nppiDivC_32sc_C3RSfs	156
7.11.2.49	nppiDivC_8u_AC4IRSfs	157
7.11.2.50	nppiDivC_8u_AC4RSfs	157
7.11.2.51	nppiDivC_8u_C1IRSfs	157
7.11.2.52	nppiDivC_8u_C1RSfs	158
7.11.2.53	nppiDivC_8u_C3IRSfs	158
7.11.2.54	nppiDivC_8u_C3RSfs	158
7.11.2.55	nppiDivC_8u_C4IRSfs	159
7.11.2.56	nppiDivC_8u_C4RSfs	159
7.12	AbsDiffC	160
7.12.1	Detailed Description	160
7.12.2	Function Documentation	160
7.12.2.1	nppiAbsDiffC_16u_C1R	160
7.12.2.2	nppiAbsDiffC_32f_C1R	160
7.12.2.3	nppiAbsDiffC_8u_C1R	161
7.13	Add	162
7.13.1	Detailed Description	167
7.13.2	Function Documentation	167
7.13.2.1	nppiAdd_16s_AC4IRSfs	167
7.13.2.2	nppiAdd_16s_AC4RSfs	167
7.13.2.3	nppiAdd_16s_C1IRSfs	168
7.13.2.4	nppiAdd_16s_C1RSfs	168
7.13.2.5	nppiAdd_16s_C3IRSfs	169
7.13.2.6	nppiAdd_16s_C3RSfs	169
7.13.2.7	nppiAdd_16s_C4IRSfs	170
7.13.2.8	nppiAdd_16s_C4RSfs	170
7.13.2.9	nppiAdd_16sc_AC4IRSfs	170
7.13.2.10	nppiAdd_16sc_AC4RSfs	171
7.13.2.11	nppiAdd_16sc_C1IRSfs	171
7.13.2.12	nppiAdd_16sc_C1RSfs	172
7.13.2.13	nppiAdd_16sc_C3IRSfs	172
7.13.2.14	nppiAdd_16sc_C3RSfs	172
7.13.2.15	nppiAdd_16u_AC4IRSfs	173
7.13.2.16	nppiAdd_16u_AC4RSfs	173

---

7.13.2.17 nppiAdd_16u_C1IRSfs . . . . .	174
7.13.2.18 nppiAdd_16u_C1RSfs . . . . .	174
7.13.2.19 nppiAdd_16u_C3IRSfs . . . . .	175
7.13.2.20 nppiAdd_16u_C3RSfs . . . . .	175
7.13.2.21 nppiAdd_16u_C4IRSfs . . . . .	175
7.13.2.22 nppiAdd_16u_C4RSfs . . . . .	176
7.13.2.23 nppiAdd_32f_AC4IR . . . . .	176
7.13.2.24 nppiAdd_32f_AC4R . . . . .	177
7.13.2.25 nppiAdd_32f_C1IR . . . . .	177
7.13.2.26 nppiAdd_32f_C1R . . . . .	177
7.13.2.27 nppiAdd_32f_C3IR . . . . .	178
7.13.2.28 nppiAdd_32f_C3R . . . . .	178
7.13.2.29 nppiAdd_32f_C4IR . . . . .	179
7.13.2.30 nppiAdd_32f_C4R . . . . .	179
7.13.2.31 nppiAdd_32fc_AC4IR . . . . .	179
7.13.2.32 nppiAdd_32fc_AC4R . . . . .	180
7.13.2.33 nppiAdd_32fc_C1IR . . . . .	180
7.13.2.34 nppiAdd_32fc_C1R . . . . .	180
7.13.2.35 nppiAdd_32fc_C3IR . . . . .	181
7.13.2.36 nppiAdd_32fc_C3R . . . . .	181
7.13.2.37 nppiAdd_32fc_C4IR . . . . .	182
7.13.2.38 nppiAdd_32fc_C4R . . . . .	182
7.13.2.39 nppiAdd_32s_C1IRSfs . . . . .	182
7.13.2.40 nppiAdd_32s_C1R . . . . .	183
7.13.2.41 nppiAdd_32s_C1RSfs . . . . .	183
7.13.2.42 nppiAdd_32s_C3IRSfs . . . . .	184
7.13.2.43 nppiAdd_32s_C3RSfs . . . . .	184
7.13.2.44 nppiAdd_32sc_AC4IRSfs . . . . .	184
7.13.2.45 nppiAdd_32sc_AC4RSfs . . . . .	185
7.13.2.46 nppiAdd_32sc_C1IRSfs . . . . .	185
7.13.2.47 nppiAdd_32sc_C1RSfs . . . . .	186
7.13.2.48 nppiAdd_32sc_C3IRSfs . . . . .	186
7.13.2.49 nppiAdd_32sc_C3RSfs . . . . .	186
7.13.2.50 nppiAdd_8u_AC4IRSfs . . . . .	187
7.13.2.51 nppiAdd_8u_AC4RSfs . . . . .	187
7.13.2.52 nppiAdd_8u_C1IRSfs . . . . .	188

7.13.2.53	nppiAdd_8u_C1RSfs	188
7.13.2.54	nppiAdd_8u_C3RSfs	189
7.13.2.55	nppiAdd_8u_C3RSfs	189
7.13.2.56	nppiAdd_8u_C4RSfs	189
7.13.2.57	nppiAdd_8u_C4RSfs	190
7.14	AddSquare	191
7.14.1	Detailed Description	191
7.14.2	Function Documentation	191
7.14.2.1	nppiAddSquare_16u32f_C1IMR	191
7.14.2.2	nppiAddSquare_16u32f_C1IR	192
7.14.2.3	nppiAddSquare_32f_C1IMR	192
7.14.2.4	nppiAddSquare_32f_C1IR	193
7.14.2.5	nppiAddSquare_8u32f_C1IMR	193
7.14.2.6	nppiAddSquare_8u32f_C1IR	193
7.15	AddProduct	194
7.15.1	Detailed Description	194
7.15.2	Function Documentation	194
7.15.2.1	nppiAddProduct_16u32f_C1IMR	194
7.15.2.2	nppiAddProduct_16u32f_C1IR	195
7.15.2.3	nppiAddProduct_32f_C1IMR	195
7.15.2.4	nppiAddProduct_32f_C1IR	196
7.15.2.5	nppiAddProduct_8u32f_C1IMR	196
7.15.2.6	nppiAddProduct_8u32f_C1IR	197
7.16	AddWeighted	198
7.16.1	Detailed Description	198
7.16.2	Function Documentation	198
7.16.2.1	nppiAddWeighted_16u32f_C1IMR	198
7.16.2.2	nppiAddWeighted_16u32f_C1IR	199
7.16.2.3	nppiAddWeighted_32f_C1IMR	199
7.16.2.4	nppiAddWeighted_32f_C1IR	200
7.16.2.5	nppiAddWeighted_8u32f_C1IMR	200
7.16.2.6	nppiAddWeighted_8u32f_C1IR	201
7.17	Mul	202
7.17.1	Detailed Description	207
7.17.2	Function Documentation	207
7.17.2.1	nppiMul_16s_AC4RSfs	207

7.17.2.2	nppiMul_16s_AC4RSfs	208
7.17.2.3	nppiMul_16s_C1RSfs	208
7.17.2.4	nppiMul_16s_C1RSfs	208
7.17.2.5	nppiMul_16s_C3RSfs	209
7.17.2.6	nppiMul_16s_C3RSfs	209
7.17.2.7	nppiMul_16s_C4RSfs	210
7.17.2.8	nppiMul_16s_C4RSfs	210
7.17.2.9	nppiMul_16sc_AC4RSfs	210
7.17.2.10	nppiMul_16sc_AC4RSfs	211
7.17.2.11	nppiMul_16sc_C1RSfs	211
7.17.2.12	nppiMul_16sc_C1RSfs	212
7.17.2.13	nppiMul_16sc_C3RSfs	212
7.17.2.14	nppiMul_16sc_C3RSfs	212
7.17.2.15	nppiMul_16u_AC4RSfs	213
7.17.2.16	nppiMul_16u_AC4RSfs	213
7.17.2.17	nppiMul_16u_C1RSfs	214
7.17.2.18	nppiMul_16u_C1RSfs	214
7.17.2.19	nppiMul_16u_C3RSfs	215
7.17.2.20	nppiMul_16u_C3RSfs	215
7.17.2.21	nppiMul_16u_C4RSfs	215
7.17.2.22	nppiMul_16u_C4RSfs	216
7.17.2.23	nppiMul_32f_AC4IR	216
7.17.2.24	nppiMul_32f_AC4R	217
7.17.2.25	nppiMul_32f_C1IR	217
7.17.2.26	nppiMul_32f_C1R	217
7.17.2.27	nppiMul_32f_C3IR	218
7.17.2.28	nppiMul_32f_C3R	218
7.17.2.29	nppiMul_32f_C4IR	219
7.17.2.30	nppiMul_32f_C4R	219
7.17.2.31	nppiMul_32fc_AC4IR	219
7.17.2.32	nppiMul_32fc_AC4R	220
7.17.2.33	nppiMul_32fc_C1IR	220
7.17.2.34	nppiMul_32fc_C1R	220
7.17.2.35	nppiMul_32fc_C3IR	221
7.17.2.36	nppiMul_32fc_C3R	221
7.17.2.37	nppiMul_32fc_C4IR	222

7.17.2.38	nppiMul_32fc_C4R	222
7.17.2.39	nppiMul_32s_C1RSfs	222
7.17.2.40	nppiMul_32s_C1R	223
7.17.2.41	nppiMul_32s_C1RSfs	223
7.17.2.42	nppiMul_32s_C3RSfs	224
7.17.2.43	nppiMul_32s_C3RSfs	224
7.17.2.44	nppiMul_32sc_AC4RSfs	224
7.17.2.45	nppiMul_32sc_AC4RSfs	225
7.17.2.46	nppiMul_32sc_C1RSfs	225
7.17.2.47	nppiMul_32sc_C1RSfs	226
7.17.2.48	nppiMul_32sc_C3RSfs	226
7.17.2.49	nppiMul_32sc_C3RSfs	226
7.17.2.50	nppiMul_8u_AC4RSfs	227
7.17.2.51	nppiMul_8u_AC4RSfs	227
7.17.2.52	nppiMul_8u_C1RSfs	228
7.17.2.53	nppiMul_8u_C1RSfs	228
7.17.2.54	nppiMul_8u_C3RSfs	229
7.17.2.55	nppiMul_8u_C3RSfs	229
7.17.2.56	nppiMul_8u_C4RSfs	229
7.17.2.57	nppiMul_8u_C4RSfs	230
7.18	MulScale	231
7.18.1	Detailed Description	232
7.18.2	Function Documentation	232
7.18.2.1	nppiMulScale_16u_AC4IR	232
7.18.2.2	nppiMulScale_16u_AC4R	233
7.18.2.3	nppiMulScale_16u_C1IR	233
7.18.2.4	nppiMulScale_16u_C1R	234
7.18.2.5	nppiMulScale_16u_C3IR	234
7.18.2.6	nppiMulScale_16u_C3R	234
7.18.2.7	nppiMulScale_16u_C4IR	235
7.18.2.8	nppiMulScale_16u_C4R	235
7.18.2.9	nppiMulScale_8u_AC4IR	236
7.18.2.10	nppiMulScale_8u_AC4R	236
7.18.2.11	nppiMulScale_8u_C1IR	236
7.18.2.12	nppiMulScale_8u_C1R	237
7.18.2.13	nppiMulScale_8u_C3IR	237

7.18.2.14	nppiMulScale_8u_C3R	238
7.18.2.15	nppiMulScale_8u_C4IR	238
7.18.2.16	nppiMulScale_8u_C4R	238
7.19	Sub	240
7.19.1	Detailed Description	245
7.19.2	Function Documentation	245
7.19.2.1	nppiSub_16s_AC4IRSfs	245
7.19.2.2	nppiSub_16s_AC4RSfs	246
7.19.2.3	nppiSub_16s_C1IRSfs	246
7.19.2.4	nppiSub_16s_C1RSfs	247
7.19.2.5	nppiSub_16s_C3IRSfs	247
7.19.2.6	nppiSub_16s_C3RSfs	247
7.19.2.7	nppiSub_16s_C4IRSfs	248
7.19.2.8	nppiSub_16s_C4RSfs	248
7.19.2.9	nppiSub_16sc_AC4IRSfs	249
7.19.2.10	nppiSub_16sc_AC4RSfs	249
7.19.2.11	nppiSub_16sc_C1IRSfs	249
7.19.2.12	nppiSub_16sc_C1RSfs	250
7.19.2.13	nppiSub_16sc_C3IRSfs	250
7.19.2.14	nppiSub_16sc_C3RSfs	251
7.19.2.15	nppiSub_16u_AC4IRSfs	251
7.19.2.16	nppiSub_16u_AC4RSfs	251
7.19.2.17	nppiSub_16u_C1IRSfs	252
7.19.2.18	nppiSub_16u_C1RSfs	252
7.19.2.19	nppiSub_16u_C3IRSfs	253
7.19.2.20	nppiSub_16u_C3RSfs	253
7.19.2.21	nppiSub_16u_C4IRSfs	254
7.19.2.22	nppiSub_16u_C4RSfs	254
7.19.2.23	nppiSub_32f_AC4IR	254
7.19.2.24	nppiSub_32f_AC4R	255
7.19.2.25	nppiSub_32f_C1IR	255
7.19.2.26	nppiSub_32f_C1R	256
7.19.2.27	nppiSub_32f_C3IR	256
7.19.2.28	nppiSub_32f_C3R	256
7.19.2.29	nppiSub_32f_C4IR	257
7.19.2.30	nppiSub_32f_C4R	257

7.19.2.31	nppiSub_32fc_AC4IR	258
7.19.2.32	nppiSub_32fc_AC4R	258
7.19.2.33	nppiSub_32fc_C1IR	258
7.19.2.34	nppiSub_32fc_C1R	259
7.19.2.35	nppiSub_32fc_C3IR	259
7.19.2.36	nppiSub_32fc_C3R	260
7.19.2.37	nppiSub_32fc_C4IR	260
7.19.2.38	nppiSub_32fc_C4R	260
7.19.2.39	nppiSub_32s_C1IRSfs	261
7.19.2.40	nppiSub_32s_C1R	261
7.19.2.41	nppiSub_32s_C1RSfs	262
7.19.2.42	nppiSub_32s_C3IRSfs	262
7.19.2.43	nppiSub_32s_C3RSfs	262
7.19.2.44	nppiSub_32s_C4IRSfs	263
7.19.2.45	nppiSub_32s_C4RSfs	263
7.19.2.46	nppiSub_32sc_AC4IRSfs	264
7.19.2.47	nppiSub_32sc_AC4RSfs	264
7.19.2.48	nppiSub_32sc_C1IRSfs	265
7.19.2.49	nppiSub_32sc_C1RSfs	265
7.19.2.50	nppiSub_32sc_C3IRSfs	265
7.19.2.51	nppiSub_32sc_C3RSfs	266
7.19.2.52	nppiSub_8u_AC4IRSfs	266
7.19.2.53	nppiSub_8u_AC4RSfs	267
7.19.2.54	nppiSub_8u_C1IRSfs	267
7.19.2.55	nppiSub_8u_C1RSfs	267
7.19.2.56	nppiSub_8u_C3IRSfs	268
7.19.2.57	nppiSub_8u_C3RSfs	268
7.19.2.58	nppiSub_8u_C4IRSfs	269
7.19.2.59	nppiSub_8u_C4RSfs	269
7.20	Div	270
7.20.1	Detailed Description	275
7.20.2	Function Documentation	275
7.20.2.1	nppiDiv_16s_AC4IRSfs	275
7.20.2.2	nppiDiv_16s_AC4RSfs	275
7.20.2.3	nppiDiv_16s_C1IRSfs	276
7.20.2.4	nppiDiv_16s_C1RSfs	276

---

7.20.2.5	nppiDiv_16s_C3IRSfs	277
7.20.2.6	nppiDiv_16s_C3RSfs	277
7.20.2.7	nppiDiv_16s_C4IRSfs	277
7.20.2.8	nppiDiv_16s_C4RSfs	278
7.20.2.9	nppiDiv_16sc_AC4IRSfs	278
7.20.2.10	nppiDiv_16sc_AC4RSfs	279
7.20.2.11	nppiDiv_16sc_C1IRSfs	279
7.20.2.12	nppiDiv_16sc_C1RSfs	279
7.20.2.13	nppiDiv_16sc_C3IRSfs	280
7.20.2.14	nppiDiv_16sc_C3RSfs	280
7.20.2.15	nppiDiv_16u_AC4IRSfs	281
7.20.2.16	nppiDiv_16u_AC4RSfs	281
7.20.2.17	nppiDiv_16u_C1IRSfs	282
7.20.2.18	nppiDiv_16u_C1RSfs	282
7.20.2.19	nppiDiv_16u_C3IRSfs	282
7.20.2.20	nppiDiv_16u_C3RSfs	283
7.20.2.21	nppiDiv_16u_C4IRSfs	283
7.20.2.22	nppiDiv_16u_C4RSfs	284
7.20.2.23	nppiDiv_32f_AC4IR	284
7.20.2.24	nppiDiv_32f_AC4R	284
7.20.2.25	nppiDiv_32f_C1IR	285
7.20.2.26	nppiDiv_32f_C1R	285
7.20.2.27	nppiDiv_32f_C3IR	286
7.20.2.28	nppiDiv_32f_C3R	286
7.20.2.29	nppiDiv_32f_C4IR	286
7.20.2.30	nppiDiv_32f_C4R	287
7.20.2.31	nppiDiv_32fc_AC4IR	287
7.20.2.32	nppiDiv_32fc_AC4R	287
7.20.2.33	nppiDiv_32fc_C1IR	288
7.20.2.34	nppiDiv_32fc_C1R	288
7.20.2.35	nppiDiv_32fc_C3IR	289
7.20.2.36	nppiDiv_32fc_C3R	289
7.20.2.37	nppiDiv_32fc_C4IR	289
7.20.2.38	nppiDiv_32fc_C4R	290
7.20.2.39	nppiDiv_32s_C1IRSfs	290
7.20.2.40	nppiDiv_32s_C1R	290

7.20.2.41	nppiDiv_32s_C1RSfs	291
7.20.2.42	nppiDiv_32s_C3IRSfs	291
7.20.2.43	nppiDiv_32s_C3RSfs	292
7.20.2.44	nppiDiv_32sc_AC4IRSfs	292
7.20.2.45	nppiDiv_32sc_AC4RSfs	292
7.20.2.46	nppiDiv_32sc_C1IRSfs	293
7.20.2.47	nppiDiv_32sc_C1RSfs	293
7.20.2.48	nppiDiv_32sc_C3IRSfs	294
7.20.2.49	nppiDiv_32sc_C3RSfs	294
7.20.2.50	nppiDiv_8u_AC4IRSfs	295
7.20.2.51	nppiDiv_8u_AC4RSfs	295
7.20.2.52	nppiDiv_8u_C1IRSfs	295
7.20.2.53	nppiDiv_8u_C1RSfs	296
7.20.2.54	nppiDiv_8u_C3IRSfs	296
7.20.2.55	nppiDiv_8u_C3RSfs	297
7.20.2.56	nppiDiv_8u_C4IRSfs	297
7.20.2.57	nppiDiv_8u_C4RSfs	297
7.21	Div_Round	299
7.21.1	Detailed Description	301
7.21.2	Function Documentation	301
7.21.2.1	nppiDiv_Round_16s_AC4IRSfs	301
7.21.2.2	nppiDiv_Round_16s_AC4RSfs	302
7.21.2.3	nppiDiv_Round_16s_C1IRSfs	302
7.21.2.4	nppiDiv_Round_16s_C1RSfs	303
7.21.2.5	nppiDiv_Round_16s_C3IRSfs	303
7.21.2.6	nppiDiv_Round_16s_C3RSfs	304
7.21.2.7	nppiDiv_Round_16s_C4IRSfs	304
7.21.2.8	nppiDiv_Round_16s_C4RSfs	305
7.21.2.9	nppiDiv_Round_16u_AC4IRSfs	305
7.21.2.10	nppiDiv_Round_16u_AC4RSfs	306
7.21.2.11	nppiDiv_Round_16u_C1IRSfs	306
7.21.2.12	nppiDiv_Round_16u_C1RSfs	307
7.21.2.13	nppiDiv_Round_16u_C3IRSfs	307
7.21.2.14	nppiDiv_Round_16u_C3RSfs	308
7.21.2.15	nppiDiv_Round_16u_C4IRSfs	308
7.21.2.16	nppiDiv_Round_16u_C4RSfs	309

7.21.2.17	nppiDiv_Round_8u_AC4IRSfs	309
7.21.2.18	nppiDiv_Round_8u_AC4RSfs	310
7.21.2.19	nppiDiv_Round_8u_C1IRSfs	310
7.21.2.20	nppiDiv_Round_8u_C1RSfs	311
7.21.2.21	nppiDiv_Round_8u_C3IRSfs	311
7.21.2.22	nppiDiv_Round_8u_C3RSfs	312
7.21.2.23	nppiDiv_Round_8u_C4IRSfs	312
7.21.2.24	nppiDiv_Round_8u_C4RSfs	313
7.22	Abs	314
7.22.1	Detailed Description	315
7.22.2	Function Documentation	315
7.22.2.1	nppiAbs_16s_AC4IR	315
7.22.2.2	nppiAbs_16s_AC4R	315
7.22.2.3	nppiAbs_16s_C1IR	316
7.22.2.4	nppiAbs_16s_C1R	316
7.22.2.5	nppiAbs_16s_C3IR	316
7.22.2.6	nppiAbs_16s_C3R	317
7.22.2.7	nppiAbs_16s_C4IR	317
7.22.2.8	nppiAbs_16s_C4R	317
7.22.2.9	nppiAbs_32f_AC4IR	318
7.22.2.10	nppiAbs_32f_AC4R	318
7.22.2.11	nppiAbs_32f_C1IR	318
7.22.2.12	nppiAbs_32f_C1R	319
7.22.2.13	nppiAbs_32f_C3IR	319
7.22.2.14	nppiAbs_32f_C3R	319
7.22.2.15	nppiAbs_32f_C4IR	320
7.22.2.16	nppiAbs_32f_C4R	320
7.23	AbsDiff	321
7.23.1	Detailed Description	321
7.23.2	Function Documentation	321
7.23.2.1	nppiAbsDiff_16u_C1R	321
7.23.2.2	nppiAbsDiff_32f_C1R	322
7.23.2.3	nppiAbsDiff_8u_C1R	322
7.23.2.4	nppiAbsDiff_8u_C3R	322
7.23.2.5	nppiAbsDiff_8u_C4R	323
7.24	Sqr	324

7.24.1	Detailed Description	326
7.24.2	Function Documentation	327
7.24.2.1	nppiSqr_16s_AC4IRSfs	327
7.24.2.2	nppiSqr_16s_AC4RSfs	327
7.24.2.3	nppiSqr_16s_C1IRSfs	327
7.24.2.4	nppiSqr_16s_C1RSfs	328
7.24.2.5	nppiSqr_16s_C3IRSfs	328
7.24.2.6	nppiSqr_16s_C3RSfs	328
7.24.2.7	nppiSqr_16s_C4IRSfs	329
7.24.2.8	nppiSqr_16s_C4RSfs	329
7.24.2.9	nppiSqr_16u_AC4IRSfs	329
7.24.2.10	nppiSqr_16u_AC4RSfs	330
7.24.2.11	nppiSqr_16u_C1IRSfs	330
7.24.2.12	nppiSqr_16u_C1RSfs	330
7.24.2.13	nppiSqr_16u_C3IRSfs	331
7.24.2.14	nppiSqr_16u_C3RSfs	331
7.24.2.15	nppiSqr_16u_C4IRSfs	331
7.24.2.16	nppiSqr_16u_C4RSfs	332
7.24.2.17	nppiSqr_32f_AC4IR	332
7.24.2.18	nppiSqr_32f_AC4R	332
7.24.2.19	nppiSqr_32f_C1IR	333
7.24.2.20	nppiSqr_32f_C1R	333
7.24.2.21	nppiSqr_32f_C3IR	333
7.24.2.22	nppiSqr_32f_C3R	334
7.24.2.23	nppiSqr_32f_C4IR	334
7.24.2.24	nppiSqr_32f_C4R	334
7.24.2.25	nppiSqr_8u_AC4IRSfs	335
7.24.2.26	nppiSqr_8u_AC4RSfs	335
7.24.2.27	nppiSqr_8u_C1IRSfs	335
7.24.2.28	nppiSqr_8u_C1RSfs	336
7.24.2.29	nppiSqr_8u_C3IRSfs	336
7.24.2.30	nppiSqr_8u_C3RSfs	336
7.24.2.31	nppiSqr_8u_C4IRSfs	337
7.24.2.32	nppiSqr_8u_C4RSfs	337
7.25	Sqrt	338
7.25.1	Detailed Description	340

7.25.2	Function Documentation	340
7.25.2.1	nppiSqrt_16s_AC4IRSfs	340
7.25.2.2	nppiSqrt_16s_AC4RSfs	341
7.25.2.3	nppiSqrt_16s_C1IRSfs	341
7.25.2.4	nppiSqrt_16s_C1RSfs	341
7.25.2.5	nppiSqrt_16s_C3IRSfs	342
7.25.2.6	nppiSqrt_16s_C3RSfs	342
7.25.2.7	nppiSqrt_16u_AC4IRSfs	342
7.25.2.8	nppiSqrt_16u_AC4RSfs	343
7.25.2.9	nppiSqrt_16u_C1IRSfs	343
7.25.2.10	nppiSqrt_16u_C1RSfs	344
7.25.2.11	nppiSqrt_16u_C3IRSfs	344
7.25.2.12	nppiSqrt_16u_C3RSfs	344
7.25.2.13	nppiSqrt_32f_AC4IR	345
7.25.2.14	nppiSqrt_32f_AC4R	345
7.25.2.15	nppiSqrt_32f_C1IR	345
7.25.2.16	nppiSqrt_32f_C1R	346
7.25.2.17	nppiSqrt_32f_C3IR	346
7.25.2.18	nppiSqrt_32f_C3R	346
7.25.2.19	nppiSqrt_32f_C4IR	347
7.25.2.20	nppiSqrt_32f_C4R	347
7.25.2.21	nppiSqrt_8u_AC4IRSfs	347
7.25.2.22	nppiSqrt_8u_AC4RSfs	348
7.25.2.23	nppiSqrt_8u_C1IRSfs	348
7.25.2.24	nppiSqrt_8u_C1RSfs	348
7.25.2.25	nppiSqrt_8u_C3IRSfs	349
7.25.2.26	nppiSqrt_8u_C3RSfs	349
7.26	Ln	350
7.26.1	Detailed Description	351
7.26.2	Function Documentation	351
7.26.2.1	nppiLn_16s_C1IRSfs	351
7.26.2.2	nppiLn_16s_C1RSfs	352
7.26.2.3	nppiLn_16s_C3IRSfs	352
7.26.2.4	nppiLn_16s_C3RSfs	352
7.26.2.5	nppiLn_16u_C1IRSfs	353
7.26.2.6	nppiLn_16u_C1RSfs	353

7.26.2.7	<code>nppiLn_16u_C3IRSfs</code>	353
7.26.2.8	<code>nppiLn_16u_C3RSfs</code>	354
7.26.2.9	<code>nppiLn_32f_C1IR</code>	354
7.26.2.10	<code>nppiLn_32f_C1R</code>	354
7.26.2.11	<code>nppiLn_32f_C3IR</code>	355
7.26.2.12	<code>nppiLn_32f_C3R</code>	355
7.26.2.13	<code>nppiLn_8u_C1IRSfs</code>	355
7.26.2.14	<code>nppiLn_8u_C1RSfs</code>	356
7.26.2.15	<code>nppiLn_8u_C3IRSfs</code>	356
7.26.2.16	<code>nppiLn_8u_C3RSfs</code>	356
7.27	Exp	357
7.27.1	Detailed Description	358
7.27.2	Function Documentation	358
7.27.2.1	<code>nppiExp_16s_C1IRSfs</code>	358
7.27.2.2	<code>nppiExp_16s_C1RSfs</code>	359
7.27.2.3	<code>nppiExp_16s_C3IRSfs</code>	359
7.27.2.4	<code>nppiExp_16s_C3RSfs</code>	359
7.27.2.5	<code>nppiExp_16u_C1IRSfs</code>	360
7.27.2.6	<code>nppiExp_16u_C1RSfs</code>	360
7.27.2.7	<code>nppiExp_16u_C3IRSfs</code>	360
7.27.2.8	<code>nppiExp_16u_C3RSfs</code>	361
7.27.2.9	<code>nppiExp_32f_C1IR</code>	361
7.27.2.10	<code>nppiExp_32f_C1R</code>	361
7.27.2.11	<code>nppiExp_32f_C3IR</code>	362
7.27.2.12	<code>nppiExp_32f_C3R</code>	362
7.27.2.13	<code>nppiExp_8u_C1IRSfs</code>	362
7.27.2.14	<code>nppiExp_8u_C1RSfs</code>	363
7.27.2.15	<code>nppiExp_8u_C3IRSfs</code>	363
7.27.2.16	<code>nppiExp_8u_C3RSfs</code>	363
7.28	Logical Operations	364
7.29	AndC	365
7.29.1	Detailed Description	367
7.29.2	Function Documentation	367
7.29.2.1	<code>nppiAndC_16u_AC4IR</code>	367
7.29.2.2	<code>nppiAndC_16u_AC4R</code>	367
7.29.2.3	<code>nppiAndC_16u_C1IR</code>	367

7.29.2.4	nppiAndC_16u_C1R	368
7.29.2.5	nppiAndC_16u_C3IR	368
7.29.2.6	nppiAndC_16u_C3R	368
7.29.2.7	nppiAndC_16u_C4IR	369
7.29.2.8	nppiAndC_16u_C4R	369
7.29.2.9	nppiAndC_32s_AC4IR	370
7.29.2.10	nppiAndC_32s_AC4R	370
7.29.2.11	nppiAndC_32s_C1IR	370
7.29.2.12	nppiAndC_32s_C1R	371
7.29.2.13	nppiAndC_32s_C3IR	371
7.29.2.14	nppiAndC_32s_C3R	371
7.29.2.15	nppiAndC_32s_C4IR	372
7.29.2.16	nppiAndC_32s_C4R	372
7.29.2.17	nppiAndC_8u_AC4IR	372
7.29.2.18	nppiAndC_8u_AC4R	373
7.29.2.19	nppiAndC_8u_C1IR	373
7.29.2.20	nppiAndC_8u_C1R	373
7.29.2.21	nppiAndC_8u_C3IR	374
7.29.2.22	nppiAndC_8u_C3R	374
7.29.2.23	nppiAndC_8u_C4IR	374
7.29.2.24	nppiAndC_8u_C4R	375
7.30	OrC	376
7.30.1	Detailed Description	378
7.30.2	Function Documentation	378
7.30.2.1	nppiOrC_16u_AC4IR	378
7.30.2.2	nppiOrC_16u_AC4R	378
7.30.2.3	nppiOrC_16u_C1IR	378
7.30.2.4	nppiOrC_16u_C1R	379
7.30.2.5	nppiOrC_16u_C3IR	379
7.30.2.6	nppiOrC_16u_C3R	379
7.30.2.7	nppiOrC_16u_C4IR	380
7.30.2.8	nppiOrC_16u_C4R	380
7.30.2.9	nppiOrC_32s_AC4IR	381
7.30.2.10	nppiOrC_32s_AC4R	381
7.30.2.11	nppiOrC_32s_C1IR	381
7.30.2.12	nppiOrC_32s_C1R	382

7.30.2.13	nppiOrC_32s_C3IR	382
7.30.2.14	nppiOrC_32s_C3R	382
7.30.2.15	nppiOrC_32s_C4IR	383
7.30.2.16	nppiOrC_32s_C4R	383
7.30.2.17	nppiOrC_8u_AC4IR	383
7.30.2.18	nppiOrC_8u_AC4R	384
7.30.2.19	nppiOrC_8u_C1IR	384
7.30.2.20	nppiOrC_8u_C1R	384
7.30.2.21	nppiOrC_8u_C3IR	385
7.30.2.22	nppiOrC_8u_C3R	385
7.30.2.23	nppiOrC_8u_C4IR	385
7.30.2.24	nppiOrC_8u_C4R	386
7.31	XorC	387
7.31.1	Detailed Description	389
7.31.2	Function Documentation	389
7.31.2.1	nppiXorC_16u_AC4IR	389
7.31.2.2	nppiXorC_16u_AC4R	389
7.31.2.3	nppiXorC_16u_C1IR	389
7.31.2.4	nppiXorC_16u_C1R	390
7.31.2.5	nppiXorC_16u_C3IR	390
7.31.2.6	nppiXorC_16u_C3R	390
7.31.2.7	nppiXorC_16u_C4IR	391
7.31.2.8	nppiXorC_16u_C4R	391
7.31.2.9	nppiXorC_32s_AC4IR	392
7.31.2.10	nppiXorC_32s_AC4R	392
7.31.2.11	nppiXorC_32s_C1IR	392
7.31.2.12	nppiXorC_32s_C1R	393
7.31.2.13	nppiXorC_32s_C3IR	393
7.31.2.14	nppiXorC_32s_C3R	393
7.31.2.15	nppiXorC_32s_C4IR	394
7.31.2.16	nppiXorC_32s_C4R	394
7.31.2.17	nppiXorC_8u_AC4IR	394
7.31.2.18	nppiXorC_8u_AC4R	395
7.31.2.19	nppiXorC_8u_C1IR	395
7.31.2.20	nppiXorC_8u_C1R	395
7.31.2.21	nppiXorC_8u_C3IR	396

7.31.2.22	nppiXorC_8u_C3R	396
7.31.2.23	nppiXorC_8u_C4IR	396
7.31.2.24	nppiXorC_8u_C4R	397
7.32	RShiftC	398
7.32.1	Detailed Description	401
7.32.2	Function Documentation	401
7.32.2.1	nppiRShiftC_16s_AC4IR	401
7.32.2.2	nppiRShiftC_16s_AC4R	401
7.32.2.3	nppiRShiftC_16s_C1IR	402
7.32.2.4	nppiRShiftC_16s_C1R	402
7.32.2.5	nppiRShiftC_16s_C3IR	402
7.32.2.6	nppiRShiftC_16s_C3R	403
7.32.2.7	nppiRShiftC_16s_C4IR	403
7.32.2.8	nppiRShiftC_16s_C4R	403
7.32.2.9	nppiRShiftC_16u_AC4IR	404
7.32.2.10	nppiRShiftC_16u_AC4R	404
7.32.2.11	nppiRShiftC_16u_C1IR	405
7.32.2.12	nppiRShiftC_16u_C1R	405
7.32.2.13	nppiRShiftC_16u_C3IR	405
7.32.2.14	nppiRShiftC_16u_C3R	406
7.32.2.15	nppiRShiftC_16u_C4IR	406
7.32.2.16	nppiRShiftC_16u_C4R	406
7.32.2.17	nppiRShiftC_32s_AC4IR	407
7.32.2.18	nppiRShiftC_32s_AC4R	407
7.32.2.19	nppiRShiftC_32s_C1IR	407
7.32.2.20	nppiRShiftC_32s_C1R	408
7.32.2.21	nppiRShiftC_32s_C3IR	408
7.32.2.22	nppiRShiftC_32s_C3R	408
7.32.2.23	nppiRShiftC_32s_C4IR	409
7.32.2.24	nppiRShiftC_32s_C4R	409
7.32.2.25	nppiRShiftC_8s_AC4IR	409
7.32.2.26	nppiRShiftC_8s_AC4R	410
7.32.2.27	nppiRShiftC_8s_C1IR	410
7.32.2.28	nppiRShiftC_8s_C1R	410
7.32.2.29	nppiRShiftC_8s_C3IR	411
7.32.2.30	nppiRShiftC_8s_C3R	411

7.32.2.31	<a href="#">nppiRShiftC_8s_C4IR</a>	411
7.32.2.32	<a href="#">nppiRShiftC_8s_C4R</a>	412
7.32.2.33	<a href="#">nppiRShiftC_8u_AC4IR</a>	412
7.32.2.34	<a href="#">nppiRShiftC_8u_AC4R</a>	412
7.32.2.35	<a href="#">nppiRShiftC_8u_C1IR</a>	413
7.32.2.36	<a href="#">nppiRShiftC_8u_C1R</a>	413
7.32.2.37	<a href="#">nppiRShiftC_8u_C3IR</a>	413
7.32.2.38	<a href="#">nppiRShiftC_8u_C3R</a>	414
7.32.2.39	<a href="#">nppiRShiftC_8u_C4IR</a>	414
7.32.2.40	<a href="#">nppiRShiftC_8u_C4R</a>	414
7.33	<a href="#">LShiftC</a>	415
7.33.1	<a href="#">Detailed Description</a>	417
7.33.2	<a href="#">Function Documentation</a>	417
7.33.2.1	<a href="#">nppiLShiftC_16u_AC4IR</a>	417
7.33.2.2	<a href="#">nppiLShiftC_16u_AC4R</a>	417
7.33.2.3	<a href="#">nppiLShiftC_16u_C1IR</a>	417
7.33.2.4	<a href="#">nppiLShiftC_16u_C1R</a>	418
7.33.2.5	<a href="#">nppiLShiftC_16u_C3IR</a>	418
7.33.2.6	<a href="#">nppiLShiftC_16u_C3R</a>	418
7.33.2.7	<a href="#">nppiLShiftC_16u_C4IR</a>	419
7.33.2.8	<a href="#">nppiLShiftC_16u_C4R</a>	419
7.33.2.9	<a href="#">nppiLShiftC_32s_AC4IR</a>	420
7.33.2.10	<a href="#">nppiLShiftC_32s_AC4R</a>	420
7.33.2.11	<a href="#">nppiLShiftC_32s_C1IR</a>	420
7.33.2.12	<a href="#">nppiLShiftC_32s_C1R</a>	421
7.33.2.13	<a href="#">nppiLShiftC_32s_C3IR</a>	421
7.33.2.14	<a href="#">nppiLShiftC_32s_C3R</a>	421
7.33.2.15	<a href="#">nppiLShiftC_32s_C4IR</a>	422
7.33.2.16	<a href="#">nppiLShiftC_32s_C4R</a>	422
7.33.2.17	<a href="#">nppiLShiftC_8u_AC4IR</a>	422
7.33.2.18	<a href="#">nppiLShiftC_8u_AC4R</a>	423
7.33.2.19	<a href="#">nppiLShiftC_8u_C1IR</a>	423
7.33.2.20	<a href="#">nppiLShiftC_8u_C1R</a>	423
7.33.2.21	<a href="#">nppiLShiftC_8u_C3IR</a>	424
7.33.2.22	<a href="#">nppiLShiftC_8u_C3R</a>	424
7.33.2.23	<a href="#">nppiLShiftC_8u_C4IR</a>	424

7.33.2.24	nppiLShiftC_8u_C4R	425
7.34	And	426
7.34.1	Detailed Description	428
7.34.2	Function Documentation	428
7.34.2.1	nppiAnd_16u_AC4IR	428
7.34.2.2	nppiAnd_16u_AC4R	428
7.34.2.3	nppiAnd_16u_C1IR	428
7.34.2.4	nppiAnd_16u_C1R	429
7.34.2.5	nppiAnd_16u_C3IR	429
7.34.2.6	nppiAnd_16u_C3R	430
7.34.2.7	nppiAnd_16u_C4IR	430
7.34.2.8	nppiAnd_16u_C4R	430
7.34.2.9	nppiAnd_32s_AC4IR	431
7.34.2.10	nppiAnd_32s_AC4R	431
7.34.2.11	nppiAnd_32s_C1IR	432
7.34.2.12	nppiAnd_32s_C1R	432
7.34.2.13	nppiAnd_32s_C3IR	432
7.34.2.14	nppiAnd_32s_C3R	433
7.34.2.15	nppiAnd_32s_C4IR	433
7.34.2.16	nppiAnd_32s_C4R	433
7.34.2.17	nppiAnd_8u_AC4IR	434
7.34.2.18	nppiAnd_8u_AC4R	434
7.34.2.19	nppiAnd_8u_C1IR	435
7.34.2.20	nppiAnd_8u_C1R	435
7.34.2.21	nppiAnd_8u_C3IR	435
7.34.2.22	nppiAnd_8u_C3R	436
7.34.2.23	nppiAnd_8u_C4IR	436
7.34.2.24	nppiAnd_8u_C4R	436
7.35	Or	438
7.35.1	Detailed Description	440
7.35.2	Function Documentation	440
7.35.2.1	nppiOr_16u_AC4IR	440
7.35.2.2	nppiOr_16u_AC4R	440
7.35.2.3	nppiOr_16u_C1IR	440
7.35.2.4	nppiOr_16u_C1R	441
7.35.2.5	nppiOr_16u_C3IR	441

---

7.35.2.6	nppiOr_16u_C3R	442
7.35.2.7	nppiOr_16u_C4IR	442
7.35.2.8	nppiOr_16u_C4R	442
7.35.2.9	nppiOr_32s_AC4IR	443
7.35.2.10	nppiOr_32s_AC4R	443
7.35.2.11	nppiOr_32s_C1IR	444
7.35.2.12	nppiOr_32s_C1R	444
7.35.2.13	nppiOr_32s_C3IR	444
7.35.2.14	nppiOr_32s_C3R	445
7.35.2.15	nppiOr_32s_C4IR	445
7.35.2.16	nppiOr_32s_C4R	445
7.35.2.17	nppiOr_8u_AC4IR	446
7.35.2.18	nppiOr_8u_AC4R	446
7.35.2.19	nppiOr_8u_C1IR	447
7.35.2.20	nppiOr_8u_C1R	447
7.35.2.21	nppiOr_8u_C3IR	447
7.35.2.22	nppiOr_8u_C3R	448
7.35.2.23	nppiOr_8u_C4IR	448
7.35.2.24	nppiOr_8u_C4R	448
7.36	Xor	450
7.36.1	Detailed Description	452
7.36.2	Function Documentation	452
7.36.2.1	nppiXor_16u_AC4IR	452
7.36.2.2	nppiXor_16u_AC4R	452
7.36.2.3	nppiXor_16u_C1IR	452
7.36.2.4	nppiXor_16u_C1R	453
7.36.2.5	nppiXor_16u_C3IR	453
7.36.2.6	nppiXor_16u_C3R	454
7.36.2.7	nppiXor_16u_C4IR	454
7.36.2.8	nppiXor_16u_C4R	454
7.36.2.9	nppiXor_32s_AC4IR	455
7.36.2.10	nppiXor_32s_AC4R	455
7.36.2.11	nppiXor_32s_C1IR	456
7.36.2.12	nppiXor_32s_C1R	456
7.36.2.13	nppiXor_32s_C3IR	456
7.36.2.14	nppiXor_32s_C3R	457

7.36.2.15	nppiXor_32s_C4IR	457
7.36.2.16	nppiXor_32s_C4R	457
7.36.2.17	nppiXor_8u_AC4IR	458
7.36.2.18	nppiXor_8u_AC4R	458
7.36.2.19	nppiXor_8u_C1IR	459
7.36.2.20	nppiXor_8u_C1R	459
7.36.2.21	nppiXor_8u_C3IR	459
7.36.2.22	nppiXor_8u_C3R	460
7.36.2.23	nppiXor_8u_C4IR	460
7.36.2.24	nppiXor_8u_C4R	460
7.37	Not	462
7.37.1	Detailed Description	462
7.37.2	Function Documentation	462
7.37.2.1	nppiNot_8u_AC4IR	462
7.37.2.2	nppiNot_8u_AC4R	463
7.37.2.3	nppiNot_8u_C1IR	463
7.37.2.4	nppiNot_8u_C1R	463
7.37.2.5	nppiNot_8u_C3IR	464
7.37.2.6	nppiNot_8u_C3R	464
7.37.2.7	nppiNot_8u_C4IR	464
7.37.2.8	nppiNot_8u_C4R	465
7.38	Alpha Composition	466
7.39	AlphaCompC	467
7.39.1	Detailed Description	468
7.39.2	Function Documentation	468
7.39.2.1	nppiAlphaCompC_16s_C1R	468
7.39.2.2	nppiAlphaCompC_16u_AC4R	469
7.39.2.3	nppiAlphaCompC_16u_C1R	469
7.39.2.4	nppiAlphaCompC_16u_C3R	470
7.39.2.5	nppiAlphaCompC_16u_C4R	470
7.39.2.6	nppiAlphaCompC_32f_C1R	471
7.39.2.7	nppiAlphaCompC_32s_C1R	471
7.39.2.8	nppiAlphaCompC_32u_C1R	472
7.39.2.9	nppiAlphaCompC_8s_C1R	472
7.39.2.10	nppiAlphaCompC_8u_AC4R	473
7.39.2.11	nppiAlphaCompC_8u_C1R	473

7.39.2.12	<code>nppiAlphaCompC_8u_C3R</code>	474
7.39.2.13	<code>nppiAlphaCompC_8u_C4R</code>	474
7.40	<code>AlphaPremulC</code>	475
7.40.1	Detailed Description	476
7.40.2	Function Documentation	476
7.40.2.1	<code>nppiAlphaPremulC_16u_AC4IR</code>	476
7.40.2.2	<code>nppiAlphaPremulC_16u_AC4R</code>	476
7.40.2.3	<code>nppiAlphaPremulC_16u_C1IR</code>	477
7.40.2.4	<code>nppiAlphaPremulC_16u_C1R</code>	477
7.40.2.5	<code>nppiAlphaPremulC_16u_C3IR</code>	478
7.40.2.6	<code>nppiAlphaPremulC_16u_C3R</code>	478
7.40.2.7	<code>nppiAlphaPremulC_16u_C4IR</code>	478
7.40.2.8	<code>nppiAlphaPremulC_16u_C4R</code>	479
7.40.2.9	<code>nppiAlphaPremulC_8u_AC4IR</code>	479
7.40.2.10	<code>nppiAlphaPremulC_8u_AC4R</code>	479
7.40.2.11	<code>nppiAlphaPremulC_8u_C1IR</code>	480
7.40.2.12	<code>nppiAlphaPremulC_8u_C1R</code>	480
7.40.2.13	<code>nppiAlphaPremulC_8u_C3IR</code>	480
7.40.2.14	<code>nppiAlphaPremulC_8u_C3R</code>	481
7.40.2.15	<code>nppiAlphaPremulC_8u_C4IR</code>	481
7.40.2.16	<code>nppiAlphaPremulC_8u_C4R</code>	481
7.41	<code>AlphaComp</code>	482
7.41.1	Detailed Description	483
7.41.2	Function Documentation	483
7.41.2.1	<code>nppiAlphaComp_16s_AC1R</code>	483
7.41.2.2	<code>nppiAlphaComp_16u_AC1R</code>	483
7.41.2.3	<code>nppiAlphaComp_16u_AC4R</code>	484
7.41.2.4	<code>nppiAlphaComp_32f_AC1R</code>	484
7.41.2.5	<code>nppiAlphaComp_32f_AC4R</code>	485
7.41.2.6	<code>nppiAlphaComp_32s_AC1R</code>	485
7.41.2.7	<code>nppiAlphaComp_32s_AC4R</code>	486
7.41.2.8	<code>nppiAlphaComp_32u_AC1R</code>	486
7.41.2.9	<code>nppiAlphaComp_32u_AC4R</code>	487
7.41.2.10	<code>nppiAlphaComp_8s_AC1R</code>	487
7.41.2.11	<code>nppiAlphaComp_8u_AC1R</code>	487
7.41.2.12	<code>nppiAlphaComp_8u_AC4R</code>	488

7.42	AlphaPremul	489
7.42.1	Detailed Description	489
7.42.2	Function Documentation	489
7.42.2.1	nppiAlphaPremul_16u_AC4IR	489
7.42.2.2	nppiAlphaPremul_16u_AC4R	490
7.42.2.3	nppiAlphaPremul_8u_AC4IR	490
7.42.2.4	nppiAlphaPremul_8u_AC4R	490
7.43	Color and Sampling Conversion	491
7.43.1	Detailed Description	491
7.44	Color Model Conversion	492
7.44.1	Detailed Description	515
7.44.2	Function Documentation	515
7.44.2.1	nppiBGRTToCbYCr422_709HDTV_8u_AC4C2R	515
7.44.2.2	nppiBGRTToCbYCr422_709HDTV_8u_C3C2R	516
7.44.2.3	nppiBGRTToCbYCr422_8u_AC4C2R	516
7.44.2.4	nppiBGRTToHLS_8u_AC4P4R	516
7.44.2.5	nppiBGRTToHLS_8u_AC4R	517
7.44.2.6	nppiBGRTToHLS_8u_AP4C4R	517
7.44.2.7	nppiBGRTToHLS_8u_AP4R	518
7.44.2.8	nppiBGRTToHLS_8u_C3P3R	518
7.44.2.9	nppiBGRTToHLS_8u_P3C3R	518
7.44.2.10	nppiBGRTToHLS_8u_P3R	519
7.44.2.11	nppiBGRTToLab_8u_C3R	519
7.44.2.12	nppiBGRTToYCbCr411_8u_AC4P3R	519
7.44.2.13	nppiBGRTToYCbCr411_8u_C3P3R	520
7.44.2.14	nppiBGRTToYCbCr420_709CSC_8u_AC4P3R	520
7.44.2.15	nppiBGRTToYCbCr420_709CSC_8u_C3P3R	520
7.44.2.16	nppiBGRTToYCbCr420_709HDTV_8u_AC4P3R	521
7.44.2.17	nppiBGRTToYCbCr420_8u_AC4P3R	521
7.44.2.18	nppiBGRTToYCbCr420_8u_C3P3R	522
7.44.2.19	nppiBGRTToYCbCr422_8u_AC4C2R	522
7.44.2.20	nppiBGRTToYCbCr422_8u_AC4P3R	522
7.44.2.21	nppiBGRTToYCbCr422_8u_C3C2R	523
7.44.2.22	nppiBGRTToYCbCr422_8u_C3P3R	523
7.44.2.23	nppiBGRTToYCrCb420_709CSC_8u_AC4P3R	524
7.44.2.24	nppiBGRTToYCrCb420_709CSC_8u_C3P3R	524

7.44.2.25 nppiBGRTToYCrCb420_8u_AC4P3R . . . . .	524
7.44.2.26 nppiBGRTToYCrCb420_8u_C3P3R . . . . .	525
7.44.2.27 nppiBGRTToYUV420_8u_AC4P3R . . . . .	525
7.44.2.28 nppiCbYCr422ToBGR_709HDTV_8u_C2C3R . . . . .	526
7.44.2.29 nppiCbYCr422ToBGR_709HDTV_8u_C2C4R . . . . .	526
7.44.2.30 nppiCbYCr422ToBGR_8u_C2C4R . . . . .	526
7.44.2.31 nppiCbYCr422ToRGB_8u_C2C3R . . . . .	527
7.44.2.32 nppiHLSToBGR_8u_AC4P4R . . . . .	527
7.44.2.33 nppiHLSToBGR_8u_AC4R . . . . .	528
7.44.2.34 nppiHLSToBGR_8u_AP4C4R . . . . .	528
7.44.2.35 nppiHLSToBGR_8u_AP4R . . . . .	528
7.44.2.36 nppiHLSToBGR_8u_C3P3R . . . . .	529
7.44.2.37 nppiHLSToBGR_8u_P3C3R . . . . .	529
7.44.2.38 nppiHLSToBGR_8u_P3R . . . . .	529
7.44.2.39 nppiHLSToRGB_8u_AC4R . . . . .	530
7.44.2.40 nppiHLSToRGB_8u_C3R . . . . .	530
7.44.2.41 nppiHSVToRGB_8u_AC4R . . . . .	530
7.44.2.42 nppiHSVToRGB_8u_C3R . . . . .	531
7.44.2.43 nppiLabToBGR_8u_C3R . . . . .	531
7.44.2.44 nppiLUVToRGB_8u_AC4R . . . . .	531
7.44.2.45 nppiLUVToRGB_8u_C3R . . . . .	532
7.44.2.46 nppiRGBToCbYCr422_8u_C3C2R . . . . .	532
7.44.2.47 nppiRGBToCbYCr422Gamma_8u_C3C2R . . . . .	532
7.44.2.48 nppiRGBToHLS_8u_AC4R . . . . .	533
7.44.2.49 nppiRGBToHLS_8u_C3R . . . . .	533
7.44.2.50 nppiRGBToHSV_8u_AC4R . . . . .	533
7.44.2.51 nppiRGBToHSV_8u_C3R . . . . .	534
7.44.2.52 nppiRGBToLUV_8u_AC4R . . . . .	534
7.44.2.53 nppiRGBToLUV_8u_C3R . . . . .	535
7.44.2.54 nppiRGBToXYZ_8u_AC4R . . . . .	535
7.44.2.55 nppiRGBToXYZ_8u_C3R . . . . .	535
7.44.2.56 nppiRGBToYCbCr420_8u_C3P3R . . . . .	536
7.44.2.57 nppiRGBToYCbCr422_8u_C3C2R . . . . .	536
7.44.2.58 nppiRGBToYCbCr422_8u_C3P3R . . . . .	536
7.44.2.59 nppiRGBToYCbCr422_8u_P3C2R . . . . .	537
7.44.2.60 nppiRGBToYCbCr_8u_AC4P3R . . . . .	537

7.44.2.61 nppiRGBToYCbCr_8u_AC4R	537
7.44.2.62 nppiRGBToYCbCr_8u_C3P3R	538
7.44.2.63 nppiRGBToYCbCr_8u_C3R	538
7.44.2.64 nppiRGBToYCbCr_8u_P3R	539
7.44.2.65 nppiRGBToYCC_8u_AC4R	539
7.44.2.66 nppiRGBToYCC_8u_C3R	539
7.44.2.67 nppiRGBToYCrCb420_8u_AC4P3R	540
7.44.2.68 nppiRGBToYCrCb422_8u_C3C2R	540
7.44.2.69 nppiRGBToYCrCb422_8u_P3C2R	540
7.44.2.70 nppiRGBToYUV420_8u_C3P3R	541
7.44.2.71 nppiRGBToYUV420_8u_P3R	541
7.44.2.72 nppiRGBToYUV422_8u_C3C2R	541
7.44.2.73 nppiRGBToYUV422_8u_C3P3R	542
7.44.2.74 nppiRGBToYUV422_8u_P3R	542
7.44.2.75 nppiRGBToYUV_8u_AC4P4R	543
7.44.2.76 nppiRGBToYUV_8u_AC4R	543
7.44.2.77 nppiRGBToYUV_8u_C3P3R	543
7.44.2.78 nppiRGBToYUV_8u_C3R	544
7.44.2.79 nppiRGBToYUV_8u_P3R	544
7.44.2.80 nppiXYZToRGB_8u_AC4R	544
7.44.2.81 nppiXYZToRGB_8u_C3R	545
7.44.2.82 nppiYCbCr411ToBGR_8u_P3C3R	545
7.44.2.83 nppiYCbCr411ToBGR_8u_P3C4R	546
7.44.2.84 nppiYCbCr420ToBGR_709CSC_8u_P3C3R	546
7.44.2.85 nppiYCbCr420ToBGR_709HDTV_8u_P3C4R	546
7.44.2.86 nppiYCbCr420ToBGR_8u_P3C3R	547
7.44.2.87 nppiYCbCr420ToBGR_8u_P3C4R	547
7.44.2.88 nppiYCbCr420ToRGB_8u_P3C3R	547
7.44.2.89 nppiYCbCr422ToBGR_8u_C2C3R	548
7.44.2.90 nppiYCbCr422ToBGR_8u_C2C4R	548
7.44.2.91 nppiYCbCr422ToBGR_8u_P3C3R	549
7.44.2.92 nppiYCbCr422ToRGB_8u_C2C3R	549
7.44.2.93 nppiYCbCr422ToRGB_8u_C2P3R	549
7.44.2.94 nppiYCbCr422ToRGB_8u_P3C3R	550
7.44.2.95 nppiYCbCrToBGR_709CSC_8u_P3C3R	550
7.44.2.96 nppiYCbCrToBGR_709CSC_8u_P3C4R	550

7.44.2.97	<code>nppiYCbCrToBGR_8u_P3C3R</code>	551
7.44.2.98	<code>nppiYCbCrToBGR_8u_P3C4R</code>	551
7.44.2.99	<code>nppiYCbCrToRGB_8u_AC4R</code>	552
7.44.2.100	<code>nppiYCbCrToRGB_8u_C3R</code>	552
7.44.2.101	<code>nppiYCbCrToRGB_8u_P3C3R</code>	552
7.44.2.102	<code>nppiYCbCrToRGB_8u_P3C4R</code>	553
7.44.2.103	<code>nppiYCbCrToRGB_8u_P3R</code>	553
7.44.2.104	<code>nppiYCCToRGB_8u_AC4R</code>	553
7.44.2.105	<code>nppiYCCToRGB_8u_C3R</code>	554
7.44.2.106	<code>nppiYCrCb420ToRGB_8u_P3C4R</code>	554
7.44.2.107	<code>nppiYCrCb422ToRGB_8u_C2C3R</code>	554
7.44.2.108	<code>nppiYCrCb422ToRGB_8u_C2P3R</code>	555
7.44.2.109	<code>nppiYUV420ToBGR_8u_P3C3R</code>	555
7.44.2.110	<code>nppiYUV420ToRGB_8u_P3AC4R</code>	555
7.44.2.111	<code>nppiYUV420ToRGB_8u_P3C3R</code>	556
7.44.2.112	<code>nppiYUV420ToRGB_8u_P3R</code>	556
7.44.2.113	<code>nppiYUV422ToRGB_8u_C2C3R</code>	556
7.44.2.114	<code>nppiYUV422ToRGB_8u_P3AC4R</code>	557
7.44.2.115	<code>nppiYUV422ToRGB_8u_P3C3R</code>	557
7.44.2.116	<code>nppiYUV422ToRGB_8u_P3R</code>	557
7.44.2.117	<code>nppiYUVToRGB_8u_AC4R</code>	558
7.44.2.118	<code>nppiYUVToRGB_8u_C3R</code>	558
7.44.2.119	<code>nppiYUVToRGB_8u_P3C3R</code>	558
7.44.2.120	<code>nppiYUVToRGB_8u_P3R</code>	559
7.45	Color Sampling Format Conversion	560
7.45.1	Detailed Description	567
7.45.2	Function Documentation	567
7.45.2.1	<code>nppiCbYCr422ToYCbCr411_8u_C2P3R</code>	567
7.45.2.2	<code>nppiCbYCr422ToYCbCr420_8u_C2P2R</code>	568
7.45.2.3	<code>nppiCbYCr422ToYCbCr420_8u_C2P3R</code>	568
7.45.2.4	<code>nppiCbYCr422ToYCbCr422_8u_C2P3R</code>	569
7.45.2.5	<code>nppiCbYCr422ToYCbCr422_8u_C2R</code>	569
7.45.2.6	<code>nppiCbYCr422ToYCrCb420_8u_C2P3R</code>	569
7.45.2.7	<code>nppiYCbCr411_8u_P2P3R</code>	570
7.45.2.8	<code>nppiYCbCr411_8u_P3P2R</code>	570
7.45.2.9	<code>nppiYCbCr411ToYCbCr420_8u_P2P3R</code>	571

7.45.2.10 nppiYCbCr411ToYCbCr420_8u_P3P2R . . . . .	571
7.45.2.11 nppiYCbCr411ToYCbCr420_8u_P3R . . . . .	571
7.45.2.12 nppiYCbCr411ToYCbCr422_8u_P2C2R . . . . .	572
7.45.2.13 nppiYCbCr411ToYCbCr422_8u_P2P3R . . . . .	572
7.45.2.14 nppiYCbCr411ToYCbCr422_8u_P3C2R . . . . .	573
7.45.2.15 nppiYCbCr411ToYCbCr422_8u_P3R . . . . .	573
7.45.2.16 nppiYCbCr411ToYCrCb420_8u_P2P3R . . . . .	573
7.45.2.17 nppiYCbCr411ToYCrCb422_8u_P3C2R . . . . .	574
7.45.2.18 nppiYCbCr411ToYCrCb422_8u_P3R . . . . .	574
7.45.2.19 nppiYCbCr420_8u_P2P3R . . . . .	575
7.45.2.20 nppiYCbCr420_8u_P3P2R . . . . .	575
7.45.2.21 nppiYCbCr420ToCbYCr422_8u_P2C2R . . . . .	575
7.45.2.22 nppiYCbCr420ToYCbCr411_8u_P2P3R . . . . .	576
7.45.2.23 nppiYCbCr420ToYCbCr411_8u_P3P2R . . . . .	576
7.45.2.24 nppiYCbCr420ToYCbCr422_8u_P2C2R . . . . .	577
7.45.2.25 nppiYCbCr420ToYCbCr422_8u_P2P3R . . . . .	577
7.45.2.26 nppiYCbCr420ToYCbCr422_8u_P3R . . . . .	578
7.45.2.27 nppiYCbCr420ToYCrCb420_8u_P2P3R . . . . .	578
7.45.2.28 nppiYCbCr422_8u_C2P3R . . . . .	578
7.45.2.29 nppiYCbCr422_8u_P3C2R . . . . .	579
7.45.2.30 nppiYCbCr422ToCbYCr422_8u_C2R . . . . .	579
7.45.2.31 nppiYCbCr422ToYCbCr411_8u_C2P2R . . . . .	580
7.45.2.32 nppiYCbCr422ToYCbCr411_8u_C2P3R . . . . .	580
7.45.2.33 nppiYCbCr422ToYCbCr411_8u_P3P2R . . . . .	580
7.45.2.34 nppiYCbCr422ToYCbCr411_8u_P3R . . . . .	581
7.45.2.35 nppiYCbCr422ToYCbCr420_8u_C2P2R . . . . .	581
7.45.2.36 nppiYCbCr422ToYCbCr420_8u_C2P3R . . . . .	582
7.45.2.37 nppiYCbCr422ToYCbCr420_8u_P3P2R . . . . .	582
7.45.2.38 nppiYCbCr422ToYCbCr420_8u_P3R . . . . .	583
7.45.2.39 nppiYCbCr422ToYCrCb420_8u_C2P3R . . . . .	583
7.45.2.40 nppiYCbCr422ToYCrCb422_8u_C2R . . . . .	583
7.45.2.41 nppiYCbCr422ToYCrCb422_8u_P3C2R . . . . .	584
7.45.2.42 nppiYCrCb420ToCbYCr422_8u_P3C2R . . . . .	584
7.45.2.43 nppiYCrCb420ToYCbCr411_8u_P3P2R . . . . .	585
7.45.2.44 nppiYCrCb420ToYCbCr420_8u_P3P2R . . . . .	585
7.45.2.45 nppiYCrCb420ToYCbCr422_8u_P3C2R . . . . .	586

7.45.2.46	nppiYCrCb420ToYCbCr422_8u_P3R	586
7.45.2.47	nppiYCrCb422ToYCbCr411_8u_C2P3R	586
7.45.2.48	nppiYCrCb422ToYCbCr420_8u_C2P3R	587
7.45.2.49	nppiYCrCb422ToYCbCr422_8u_C2P3R	587
7.46	Color Gamma Correction	588
7.46.1	Detailed Description	589
7.46.2	Function Documentation	589
7.46.2.1	nppiGammaFwd_8u_AC4IR	589
7.46.2.2	nppiGammaFwd_8u_AC4R	589
7.46.2.3	nppiGammaFwd_8u_C3IR	590
7.46.2.4	nppiGammaFwd_8u_C3R	590
7.46.2.5	nppiGammaFwd_8u_IP3R	590
7.46.2.6	nppiGammaFwd_8u_P3R	591
7.46.2.7	nppiGammaInv_8u_AC4IR	591
7.46.2.8	nppiGammaInv_8u_AC4R	591
7.46.2.9	nppiGammaInv_8u_C3IR	592
7.46.2.10	nppiGammaInv_8u_C3R	592
7.46.2.11	nppiGammaInv_8u_IP3R	592
7.46.2.12	nppiGammaInv_8u_P3R	593
7.47	Complement Color Key	594
7.47.1	Detailed Description	594
7.47.2	Function Documentation	594
7.47.2.1	nppiAlphaCompColorKey_8u_AC4R	594
7.47.2.2	nppiCompColorKey_8u_C1R	595
7.47.2.3	nppiCompColorKey_8u_C3R	595
7.47.2.4	nppiCompColorKey_8u_C4R	596
7.48	Color Processing	597
7.48.1	Detailed Description	597
7.48.2	Function Documentation	598
7.48.2.1	nppiColorTwist32f_8u_AC4R	598
7.48.2.2	nppiColorTwist32f_8u_C3R	598
7.48.2.3	nppiColorTwist32f_8u_P3R	598
7.48.2.4	nppiLUT_Linear_8u_AC4R	599
7.48.2.5	nppiLUT_Linear_8u_C1R	600
7.48.2.6	nppiLUT_Linear_8u_C3R	600
7.48.2.7	nppiLUT_Linear_8u_C4R	601

7.49	Compression	602
7.49.1	Detailed Description	602
7.50	Quantization Functions	603
7.50.1	Function Documentation	603
7.50.1.1	nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R	603
7.50.1.2	nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R	604
7.50.1.3	nppiQuantFwdRawTableInit_JPEG_8u	604
7.50.1.4	nppiQuantFwdTableInit_JPEG_8u16u	604
7.50.1.5	nppiQuantInvTableInit_JPEG_8u16u	605
7.51	Labeling and Segmentation	606
7.51.1	Detailed Description	606
7.51.2	Typedef Documentation	606
7.51.2.1	NppiGraphcutState	606
7.52	GraphCut	607
7.52.1	Function Documentation	608
7.52.1.1	nppiGraphcut8_32f8u	608
7.52.1.2	nppiGraphcut8_32s8u	609
7.52.1.3	nppiGraphcut8GetSize	609
7.52.1.4	nppiGraphcut8InitAlloc	610
7.52.1.5	nppiGraphcut_32f8u	610
7.52.1.6	nppiGraphcut_32s8u	611
7.52.1.7	nppiGraphcutFree	612
7.52.1.8	nppiGraphcutGetSize	612
7.52.1.9	nppiGraphcutInitAlloc	613
7.53	Data Exchange and Initialization	614
7.53.1	Detailed Description	614
7.54	Set	615
7.54.1	Function Documentation	619
7.54.1.1	nppiSet_16s_AC4MR	619
7.54.1.2	nppiSet_16s_AC4R	619
7.54.1.3	nppiSet_16s_C1MR	620
7.54.1.4	nppiSet_16s_C1R	620
7.54.1.5	nppiSet_16s_C2R	621
7.54.1.6	nppiSet_16s_C4CR	621
7.54.1.7	nppiSet_16s_C4MR	621
7.54.1.8	nppiSet_16s_C4R	622

---

7.54.1.9	nppiSet_16sc_AC4R	622
7.54.1.10	nppiSet_16sc_C1R	622
7.54.1.11	nppiSet_16sc_C2R	623
7.54.1.12	nppiSet_16sc_C3R	623
7.54.1.13	nppiSet_16sc_C4R	623
7.54.1.14	nppiSet_16u_AC4MR	624
7.54.1.15	nppiSet_16u_AC4R	624
7.54.1.16	nppiSet_16u_C1MR	624
7.54.1.17	nppiSet_16u_C1R	625
7.54.1.18	nppiSet_16u_C2R	625
7.54.1.19	nppiSet_16u_C4CR	625
7.54.1.20	nppiSet_16u_C4MR	626
7.54.1.21	nppiSet_16u_C4R	626
7.54.1.22	nppiSet_32f_AC4MR	627
7.54.1.23	nppiSet_32f_AC4R	627
7.54.1.24	nppiSet_32f_C1MR	627
7.54.1.25	nppiSet_32f_C1R	628
7.54.1.26	nppiSet_32f_C4CR	628
7.54.1.27	nppiSet_32f_C4MR	628
7.54.1.28	nppiSet_32f_C4R	629
7.54.1.29	nppiSet_32fc_AC4R	629
7.54.1.30	nppiSet_32fc_C1R	630
7.54.1.31	nppiSet_32fc_C2R	630
7.54.1.32	nppiSet_32fc_C3R	630
7.54.1.33	nppiSet_32fc_C4R	631
7.54.1.34	nppiSet_32s_AC4MR	631
7.54.1.35	nppiSet_32s_AC4R	631
7.54.1.36	nppiSet_32s_C1MR	632
7.54.1.37	nppiSet_32s_C1R	632
7.54.1.38	nppiSet_32s_C4CR	632
7.54.1.39	nppiSet_32s_C4MR	633
7.54.1.40	nppiSet_32s_C4R	633
7.54.1.41	nppiSet_32sc_AC4R	633
7.54.1.42	nppiSet_32sc_C1R	634
7.54.1.43	nppiSet_32sc_C2R	634
7.54.1.44	nppiSet_32sc_C3R	634

7.54.1.45	nppiSet_32sc_C4R	635
7.54.1.46	nppiSet_8s_AC4R	635
7.54.1.47	nppiSet_8s_C1R	635
7.54.1.48	nppiSet_8s_C2R	636
7.54.1.49	nppiSet_8s_C3R	636
7.54.1.50	nppiSet_8s_C4R	636
7.54.1.51	nppiSet_8u_AC4MR	637
7.54.1.52	nppiSet_8u_AC4R	637
7.54.1.53	nppiSet_8u_C1MR	637
7.54.1.54	nppiSet_8u_C1R	638
7.54.1.55	nppiSet_8u_C4CR	638
7.54.1.56	nppiSet_8u_C4MR	639
7.54.1.57	nppiSet_8u_C4R	639
7.55	Copy	640
7.55.1	Function Documentation	648
7.55.1.1	nppiCopy_16s_AC4MR	648
7.55.1.2	nppiCopy_16s_AC4R	649
7.55.1.3	nppiCopy_16s_C1C3R	649
7.55.1.4	nppiCopy_16s_C1C4R	650
7.55.1.5	nppiCopy_16s_C1MR	650
7.55.1.6	nppiCopy_16s_C1R	650
7.55.1.7	nppiCopy_16s_C3C1R	651
7.55.1.8	nppiCopy_16s_C3CR	651
7.55.1.9	nppiCopy_16s_C3MR	651
7.55.1.10	nppiCopy_16s_C3P3R	652
7.55.1.11	nppiCopy_16s_C4C1R	652
7.55.1.12	nppiCopy_16s_C4CR	652
7.55.1.13	nppiCopy_16s_C4MR	653
7.55.1.14	nppiCopy_16s_C4P4R	653
7.55.1.15	nppiCopy_16s_C4R	653
7.55.1.16	nppiCopy_16s_P3C3R	654
7.55.1.17	nppiCopy_16s_P4C4R	654
7.55.1.18	nppiCopy_16sc_AC4R	654
7.55.1.19	nppiCopy_16sc_C1R	655
7.55.1.20	nppiCopy_16sc_C2R	655
7.55.1.21	nppiCopy_16sc_C3R	655

---

7.55.1.22 nppiCopy_16sc_C4R . . . . .	656
7.55.1.23 nppiCopy_16u_AC4MR . . . . .	656
7.55.1.24 nppiCopy_16u_AC4R . . . . .	656
7.55.1.25 nppiCopy_16u_C1C3R . . . . .	657
7.55.1.26 nppiCopy_16u_C1C4R . . . . .	657
7.55.1.27 nppiCopy_16u_C1MR . . . . .	657
7.55.1.28 nppiCopy_16u_C1R . . . . .	658
7.55.1.29 nppiCopy_16u_C3C1R . . . . .	658
7.55.1.30 nppiCopy_16u_C3CR . . . . .	658
7.55.1.31 nppiCopy_16u_C3MR . . . . .	659
7.55.1.32 nppiCopy_16u_C3P3R . . . . .	659
7.55.1.33 nppiCopy_16u_C4C1R . . . . .	659
7.55.1.34 nppiCopy_16u_C4CR . . . . .	660
7.55.1.35 nppiCopy_16u_C4MR . . . . .	660
7.55.1.36 nppiCopy_16u_C4P4R . . . . .	660
7.55.1.37 nppiCopy_16u_C4R . . . . .	661
7.55.1.38 nppiCopy_16u_P3C3R . . . . .	661
7.55.1.39 nppiCopy_16u_P4C4R . . . . .	661
7.55.1.40 nppiCopy_32f_AC4MR . . . . .	662
7.55.1.41 nppiCopy_32f_AC4R . . . . .	662
7.55.1.42 nppiCopy_32f_C1C3R . . . . .	662
7.55.1.43 nppiCopy_32f_C1C4R . . . . .	663
7.55.1.44 nppiCopy_32f_C1MR . . . . .	663
7.55.1.45 nppiCopy_32f_C1R . . . . .	663
7.55.1.46 nppiCopy_32f_C3C1R . . . . .	664
7.55.1.47 nppiCopy_32f_C3CR . . . . .	664
7.55.1.48 nppiCopy_32f_C3MR . . . . .	664
7.55.1.49 nppiCopy_32f_C3P3R . . . . .	665
7.55.1.50 nppiCopy_32f_C4C1R . . . . .	665
7.55.1.51 nppiCopy_32f_C4CR . . . . .	665
7.55.1.52 nppiCopy_32f_C4MR . . . . .	666
7.55.1.53 nppiCopy_32f_C4P4R . . . . .	666
7.55.1.54 nppiCopy_32f_C4R . . . . .	666
7.55.1.55 nppiCopy_32f_P3C3R . . . . .	667
7.55.1.56 nppiCopy_32f_P4C4R . . . . .	667
7.55.1.57 nppiCopy_32fc_AC4R . . . . .	667

---

7.55.1.58 nppiCopy_32fc_C1R . . . . .	668
7.55.1.59 nppiCopy_32fc_C2R . . . . .	668
7.55.1.60 nppiCopy_32fc_C3R . . . . .	668
7.55.1.61 nppiCopy_32fc_C4R . . . . .	669
7.55.1.62 nppiCopy_32s_AC4MR . . . . .	669
7.55.1.63 nppiCopy_32s_AC4R . . . . .	669
7.55.1.64 nppiCopy_32s_C1C3R . . . . .	670
7.55.1.65 nppiCopy_32s_C1C4R . . . . .	670
7.55.1.66 nppiCopy_32s_C1MR . . . . .	670
7.55.1.67 nppiCopy_32s_C1R . . . . .	671
7.55.1.68 nppiCopy_32s_C3C1R . . . . .	671
7.55.1.69 nppiCopy_32s_C3CR . . . . .	671
7.55.1.70 nppiCopy_32s_C3MR . . . . .	672
7.55.1.71 nppiCopy_32s_C3P3R . . . . .	672
7.55.1.72 nppiCopy_32s_C4C1R . . . . .	672
7.55.1.73 nppiCopy_32s_C4CR . . . . .	673
7.55.1.74 nppiCopy_32s_C4MR . . . . .	673
7.55.1.75 nppiCopy_32s_C4P4R . . . . .	673
7.55.1.76 nppiCopy_32s_C4R . . . . .	674
7.55.1.77 nppiCopy_32s_P3C3R . . . . .	674
7.55.1.78 nppiCopy_32s_P4C4R . . . . .	674
7.55.1.79 nppiCopy_32sc_AC4R . . . . .	675
7.55.1.80 nppiCopy_32sc_C1R . . . . .	675
7.55.1.81 nppiCopy_32sc_C2R . . . . .	675
7.55.1.82 nppiCopy_32sc_C3R . . . . .	676
7.55.1.83 nppiCopy_32sc_C4R . . . . .	676
7.55.1.84 nppiCopy_8s_AC4R . . . . .	676
7.55.1.85 nppiCopy_8s_C1R . . . . .	677
7.55.1.86 nppiCopy_8s_C2R . . . . .	677
7.55.1.87 nppiCopy_8s_C3R . . . . .	677
7.55.1.88 nppiCopy_8s_C4R . . . . .	678
7.55.1.89 nppiCopy_8u_AC4MR . . . . .	678
7.55.1.90 nppiCopy_8u_AC4R . . . . .	678
7.55.1.91 nppiCopy_8u_C1C3R . . . . .	679
7.55.1.92 nppiCopy_8u_C1C4R . . . . .	679
7.55.1.93 nppiCopy_8u_C1MR . . . . .	679

7.55.1.94	nppiCopy_8u_C1R	680
7.55.1.95	nppiCopy_8u_C3C1R	680
7.55.1.96	nppiCopy_8u_C3CR	680
7.55.1.97	nppiCopy_8u_C3MR	681
7.55.1.98	nppiCopy_8u_C3P3R	681
7.55.1.99	nppiCopy_8u_C4C1R	681
7.55.1.100	nppiCopy_8u_C4CR	682
7.55.1.101	nppiCopy_8u_C4MR	682
7.55.1.102	nppiCopy_8u_C4P4R	682
7.55.1.103	nppiCopy_8u_C4R	683
7.55.1.104	nppiCopy_8u_P3C3R	683
7.55.1.105	nppiCopy_8u_P4C4R	683
7.56	Convert	684
7.56.1	Function Documentation	691
7.56.1.1	nppiConvert_16s16u_C1Rs	691
7.56.1.2	nppiConvert_16s32f_AC4R	692
7.56.1.3	nppiConvert_16s32f_C1R	692
7.56.1.4	nppiConvert_16s32f_C3R	692
7.56.1.5	nppiConvert_16s32f_C4R	693
7.56.1.6	nppiConvert_16s32s_AC4R	693
7.56.1.7	nppiConvert_16s32s_C1R	693
7.56.1.8	nppiConvert_16s32s_C3R	694
7.56.1.9	nppiConvert_16s32s_C4R	694
7.56.1.10	nppiConvert_16s32u_C1Rs	694
7.56.1.11	nppiConvert_16s8s_C1RSfs	695
7.56.1.12	nppiConvert_16s8u_AC4R	695
7.56.1.13	nppiConvert_16s8u_C1R	695
7.56.1.14	nppiConvert_16s8u_C3R	695
7.56.1.15	nppiConvert_16s8u_C4R	696
7.56.1.16	nppiConvert_16u16s_C1RSfs	696
7.56.1.17	nppiConvert_16u32f_AC4R	696
7.56.1.18	nppiConvert_16u32f_C1R	696
7.56.1.19	nppiConvert_16u32f_C3R	697
7.56.1.20	nppiConvert_16u32f_C4R	697
7.56.1.21	nppiConvert_16u32s_AC4R	698
7.56.1.22	nppiConvert_16u32s_C1R	698

---

7.56.1.23 nppiConvert_16u32s_C3R	698
7.56.1.24 nppiConvert_16u32s_C4R	699
7.56.1.25 nppiConvert_16u32u_C1R	699
7.56.1.26 nppiConvert_16u8s_C1RSfs	699
7.56.1.27 nppiConvert_16u8u_AC4R	699
7.56.1.28 nppiConvert_16u8u_C1R	700
7.56.1.29 nppiConvert_16u8u_C3R	700
7.56.1.30 nppiConvert_16u8u_C4R	700
7.56.1.31 nppiConvert_32f16s_AC4R	701
7.56.1.32 nppiConvert_32f16s_C1R	701
7.56.1.33 nppiConvert_32f16s_C1RSfs	702
7.56.1.34 nppiConvert_32f16s_C3R	702
7.56.1.35 nppiConvert_32f16s_C4R	702
7.56.1.36 nppiConvert_32f16u_AC4R	703
7.56.1.37 nppiConvert_32f16u_C1R	703
7.56.1.38 nppiConvert_32f16u_C1RSfs	704
7.56.1.39 nppiConvert_32f16u_C3R	704
7.56.1.40 nppiConvert_32f16u_C4R	704
7.56.1.41 nppiConvert_32f32s_C1RSfs	705
7.56.1.42 nppiConvert_32f32u_C1RSfs	705
7.56.1.43 nppiConvert_32f8s_AC4R	706
7.56.1.44 nppiConvert_32f8s_C1R	706
7.56.1.45 nppiConvert_32f8s_C1RSfs	706
7.56.1.46 nppiConvert_32f8s_C3R	707
7.56.1.47 nppiConvert_32f8s_C4R	707
7.56.1.48 nppiConvert_32f8u_AC4R	708
7.56.1.49 nppiConvert_32f8u_C1R	708
7.56.1.50 nppiConvert_32f8u_C1RSfs	708
7.56.1.51 nppiConvert_32f8u_C3R	709
7.56.1.52 nppiConvert_32f8u_C4R	709
7.56.1.53 nppiConvert_32s16s_C1RSfs	710
7.56.1.54 nppiConvert_32s16u_C1RSfs	710
7.56.1.55 nppiConvert_32s32f_C1R	710
7.56.1.56 nppiConvert_32s32u_C1Rs	710
7.56.1.57 nppiConvert_32s8s_AC4R	710
7.56.1.58 nppiConvert_32s8s_C1R	711

---

7.56.1.59 nppiConvert_32s8s_C3R	711
7.56.1.60 nppiConvert_32s8s_C4R	712
7.56.1.61 nppiConvert_32s8u_AC4R	712
7.56.1.62 nppiConvert_32s8u_C1R	712
7.56.1.63 nppiConvert_32s8u_C3R	713
7.56.1.64 nppiConvert_32s8u_C4R	713
7.56.1.65 nppiConvert_32u16s_C1RSfs	713
7.56.1.66 nppiConvert_32u16u_C1RSfs	713
7.56.1.67 nppiConvert_32u32f_C1R	713
7.56.1.68 nppiConvert_32u32s_C1RSfs	714
7.56.1.69 nppiConvert_32u8s_C1RSfs	714
7.56.1.70 nppiConvert_32u8u_C1RSfs	714
7.56.1.71 nppiConvert_8s16s_C1R	714
7.56.1.72 nppiConvert_8s16u_C1Rs	714
7.56.1.73 nppiConvert_8s32f_AC4R	715
7.56.1.74 nppiConvert_8s32f_C1R	715
7.56.1.75 nppiConvert_8s32f_C3R	715
7.56.1.76 nppiConvert_8s32f_C4R	716
7.56.1.77 nppiConvert_8s32s_AC4R	716
7.56.1.78 nppiConvert_8s32s_C1R	716
7.56.1.79 nppiConvert_8s32s_C3R	717
7.56.1.80 nppiConvert_8s32s_C4R	717
7.56.1.81 nppiConvert_8s32u_C1Rs	718
7.56.1.82 nppiConvert_8s8u_C1Rs	718
7.56.1.83 nppiConvert_8u16s_AC4R	718
7.56.1.84 nppiConvert_8u16s_C1R	719
7.56.1.85 nppiConvert_8u16s_C3R	719
7.56.1.86 nppiConvert_8u16s_C4R	719
7.56.1.87 nppiConvert_8u16u_AC4R	720
7.56.1.88 nppiConvert_8u16u_C1R	720
7.56.1.89 nppiConvert_8u16u_C3R	720
7.56.1.90 nppiConvert_8u16u_C4R	721
7.56.1.91 nppiConvert_8u32f_AC4R	721
7.56.1.92 nppiConvert_8u32f_C1R	721
7.56.1.93 nppiConvert_8u32f_C3R	722
7.56.1.94 nppiConvert_8u32f_C4R	722

7.56.1.95	nppiConvert_8u32s_AC4R	722
7.56.1.96	nppiConvert_8u32s_C1R	723
7.56.1.97	nppiConvert_8u32s_C3R	723
7.56.1.98	nppiConvert_8u32s_C4R	723
7.56.1.99	nppiConvert_8u8s_C1RSfs	724
7.57	Copy Constant Border	725
7.57.1	Function Documentation	725
7.57.1.1	nppiCopyConstBorder_32s_C1R	725
7.57.1.2	nppiCopyConstBorder_8u_AC4R	726
7.57.1.3	nppiCopyConstBorder_8u_C1R	726
7.57.1.4	nppiCopyConstBorder_8u_C4R	727
7.58	Transpose And Swap Channels	728
7.58.1	Function Documentation	728
7.58.1.1	nppiSwapChannels_8u_C4IR	728
7.58.1.2	nppiTranspose_8u_C1R	728
7.59	Filtering Functions	730
7.59.1	Detailed Description	730
7.60	1D Linear Filter	731
7.60.1	Function Documentation	731
7.60.1.1	nppiFilterColumn_8u_C1R	731
7.60.1.2	nppiFilterColumn_8u_C4R	732
7.60.1.3	nppiFilterRow_8u_C1R	732
7.60.1.4	nppiFilterRow_8u_C4R	733
7.61	1D Window Sum	734
7.61.1	Function Documentation	734
7.61.1.1	nppiSumWindowColumn_8u32f_C1R	734
7.61.1.2	nppiSumWindowRow_8u32f_C1R	734
7.62	Convolution	736
7.62.1	Function Documentation	736
7.62.1.1	nppiFilter_8u_C1R	736
7.62.1.2	nppiFilter_8u_C4R	737
7.63	2D Fixed Linear Filters	738
7.63.1	Function Documentation	738
7.63.1.1	nppiFilterBox_8u_C1R	738
7.63.1.2	nppiFilterBox_8u_C4R	738
7.64	Rank Filters	740

---

7.64.1	Function Documentation	740
7.64.1.1	nppiFilterMax_8u_C1R	740
7.64.1.2	nppiFilterMax_8u_C4R	741
7.64.1.3	nppiFilterMin_8u_C1R	741
7.64.1.4	nppiFilterMin_8u_C4R	741
7.65	Geometry Transforms	743
7.65.1	Detailed Description	743
7.65.2	Geometric Transform API Specifics	743
7.65.2.1	Geometric Transforms and ROIs	743
7.65.2.2	Pixel Interpolation	743
7.66	Resize	744
7.66.1	Detailed Description	744
7.66.2	Error Codes	744
7.66.3	Function Documentation	744
7.66.3.1	nppiResize_8u_C1R	744
7.66.3.2	nppiResize_8u_C4R	745
7.67	Rotate	746
7.67.1	Detailed Description	747
7.67.2	Rotate Error Codes	747
7.67.3	Function Documentation	747
7.67.3.1	nppiGetRotateBound	747
7.67.3.2	nppiGetRotateQuad	748
7.67.3.3	nppiRotate_16u_AC4R	748
7.67.3.4	nppiRotate_16u_C1R	749
7.67.3.5	nppiRotate_16u_C3R	749
7.67.3.6	nppiRotate_16u_C4R	750
7.67.3.7	nppiRotate_32f_AC4R	750
7.67.3.8	nppiRotate_32f_C1R	751
7.67.3.9	nppiRotate_32f_C3R	751
7.67.3.10	nppiRotate_32f_C4R	752
7.67.3.11	nppiRotate_8u_AC4R	752
7.67.3.12	nppiRotate_8u_C1R	753
7.67.3.13	nppiRotate_8u_C3R	753
7.67.3.14	nppiRotate_8u_C4R	754
7.68	Mirror	755
7.68.1	Detailed Description	756

---

---

7.68.2	Mirror Error Codes . . . . .	756
7.68.3	Function Documentation . . . . .	756
7.68.3.1	nppiMirror_16u_AC4R . . . . .	756
7.68.3.2	nppiMirror_16u_C1R . . . . .	757
7.68.3.3	nppiMirror_16u_C3R . . . . .	757
7.68.3.4	nppiMirror_16u_C4R . . . . .	757
7.68.3.5	nppiMirror_32f_AC4R . . . . .	758
7.68.3.6	nppiMirror_32f_C1R . . . . .	758
7.68.3.7	nppiMirror_32f_C3R . . . . .	758
7.68.3.8	nppiMirror_32f_C4R . . . . .	759
7.68.3.9	nppiMirror_32s_AC4R . . . . .	759
7.68.3.10	nppiMirror_32s_C1R . . . . .	759
7.68.3.11	nppiMirror_32s_C3R . . . . .	760
7.68.3.12	nppiMirror_32s_C4R . . . . .	760
7.68.3.13	nppiMirror_8u_AC4R . . . . .	760
7.68.3.14	nppiMirror_8u_C1R . . . . .	761
7.68.3.15	nppiMirror_8u_C3R . . . . .	761
7.68.3.16	nppiMirror_8u_C4R . . . . .	761
7.69	Affine Transforms . . . . .	762
7.69.1	Detailed Description . . . . .	771
7.69.2	Affine Transform Error Codes . . . . .	771
7.69.3	Function Documentation . . . . .	771
7.69.3.1	nppiGetAffineBound . . . . .	771
7.69.3.2	nppiGetAffineQuad . . . . .	771
7.69.3.3	nppiGetAffineTransform . . . . .	772
7.69.3.4	nppiWarpAffine_16u_AC4R . . . . .	773
7.69.3.5	nppiWarpAffine_16u_C1R . . . . .	773
7.69.3.6	nppiWarpAffine_16u_C3R . . . . .	774
7.69.3.7	nppiWarpAffine_16u_C4R . . . . .	774
7.69.3.8	nppiWarpAffine_16u_P3R . . . . .	775
7.69.3.9	nppiWarpAffine_16u_P4R . . . . .	775
7.69.3.10	nppiWarpAffine_32f_AC4R . . . . .	776
7.69.3.11	nppiWarpAffine_32f_C1R . . . . .	776
7.69.3.12	nppiWarpAffine_32f_C3R . . . . .	777
7.69.3.13	nppiWarpAffine_32f_C4R . . . . .	777
7.69.3.14	nppiWarpAffine_32f_P3R . . . . .	778

---

7.69.3.15	nppiWarpAffine_32f_P4R	778
7.69.3.16	nppiWarpAffine_32s_AC4R	779
7.69.3.17	nppiWarpAffine_32s_C1R	779
7.69.3.18	nppiWarpAffine_32s_C3R	780
7.69.3.19	nppiWarpAffine_32s_C4R	780
7.69.3.20	nppiWarpAffine_32s_P3R	781
7.69.3.21	nppiWarpAffine_32s_P4R	781
7.69.3.22	nppiWarpAffine_64f_AC4R	782
7.69.3.23	nppiWarpAffine_64f_C1R	782
7.69.3.24	nppiWarpAffine_64f_C3R	783
7.69.3.25	nppiWarpAffine_64f_C4R	783
7.69.3.26	nppiWarpAffine_64f_P3R	784
7.69.3.27	nppiWarpAffine_64f_P4R	784
7.69.3.28	nppiWarpAffine_8u_AC4R	785
7.69.3.29	nppiWarpAffine_8u_C1R	785
7.69.3.30	nppiWarpAffine_8u_C3R	786
7.69.3.31	nppiWarpAffine_8u_C4R	786
7.69.3.32	nppiWarpAffine_8u_P3R	787
7.69.3.33	nppiWarpAffine_8u_P4R	787
7.69.3.34	nppiWarpAffineBack_16u_AC4R	788
7.69.3.35	nppiWarpAffineBack_16u_C1R	788
7.69.3.36	nppiWarpAffineBack_16u_C3R	789
7.69.3.37	nppiWarpAffineBack_16u_C4R	789
7.69.3.38	nppiWarpAffineBack_16u_P3R	790
7.69.3.39	nppiWarpAffineBack_16u_P4R	790
7.69.3.40	nppiWarpAffineBack_32f_AC4R	791
7.69.3.41	nppiWarpAffineBack_32f_C1R	791
7.69.3.42	nppiWarpAffineBack_32f_C3R	792
7.69.3.43	nppiWarpAffineBack_32f_C4R	792
7.69.3.44	nppiWarpAffineBack_32f_P3R	793
7.69.3.45	nppiWarpAffineBack_32f_P4R	793
7.69.3.46	nppiWarpAffineBack_32s_AC4R	794
7.69.3.47	nppiWarpAffineBack_32s_C1R	794
7.69.3.48	nppiWarpAffineBack_32s_C3R	795
7.69.3.49	nppiWarpAffineBack_32s_C4R	795
7.69.3.50	nppiWarpAffineBack_32s_P3R	796

7.69.3.51	<code>nppiWarpAffineBack_32s_P4R</code>	796
7.69.3.52	<code>nppiWarpAffineBack_8u_AC4R</code>	797
7.69.3.53	<code>nppiWarpAffineBack_8u_C1R</code>	797
7.69.3.54	<code>nppiWarpAffineBack_8u_C3R</code>	798
7.69.3.55	<code>nppiWarpAffineBack_8u_C4R</code>	798
7.69.3.56	<code>nppiWarpAffineBack_8u_P3R</code>	799
7.69.3.57	<code>nppiWarpAffineBack_8u_P4R</code>	799
7.69.3.58	<code>nppiWarpAffineQuad_16u_AC4R</code>	800
7.69.3.59	<code>nppiWarpAffineQuad_16u_C1R</code>	800
7.69.3.60	<code>nppiWarpAffineQuad_16u_C3R</code>	801
7.69.3.61	<code>nppiWarpAffineQuad_16u_C4R</code>	801
7.69.3.62	<code>nppiWarpAffineQuad_16u_P3R</code>	802
7.69.3.63	<code>nppiWarpAffineQuad_16u_P4R</code>	802
7.69.3.64	<code>nppiWarpAffineQuad_32f_AC4R</code>	803
7.69.3.65	<code>nppiWarpAffineQuad_32f_C1R</code>	803
7.69.3.66	<code>nppiWarpAffineQuad_32f_C3R</code>	804
7.69.3.67	<code>nppiWarpAffineQuad_32f_C4R</code>	804
7.69.3.68	<code>nppiWarpAffineQuad_32f_P3R</code>	805
7.69.3.69	<code>nppiWarpAffineQuad_32f_P4R</code>	805
7.69.3.70	<code>nppiWarpAffineQuad_32s_AC4R</code>	806
7.69.3.71	<code>nppiWarpAffineQuad_32s_C1R</code>	806
7.69.3.72	<code>nppiWarpAffineQuad_32s_C3R</code>	807
7.69.3.73	<code>nppiWarpAffineQuad_32s_C4R</code>	807
7.69.3.74	<code>nppiWarpAffineQuad_32s_P3R</code>	808
7.69.3.75	<code>nppiWarpAffineQuad_32s_P4R</code>	808
7.69.3.76	<code>nppiWarpAffineQuad_8u_AC4R</code>	809
7.69.3.77	<code>nppiWarpAffineQuad_8u_C1R</code>	809
7.69.3.78	<code>nppiWarpAffineQuad_8u_C3R</code>	810
7.69.3.79	<code>nppiWarpAffineQuad_8u_C4R</code>	810
7.69.3.80	<code>nppiWarpAffineQuad_8u_P3R</code>	811
7.69.3.81	<code>nppiWarpAffineQuad_8u_P4R</code>	811
7.70	Perspective Transform	812
7.70.1	Detailed Description	820
7.70.2	Perspective Transform Error Codes	820
7.70.3	Function Documentation	820
7.70.3.1	<code>nppiGetPerspectiveBound</code>	820

---

7.70.3.2	<a href="#">nppiGetPerspectiveQuad</a>	821
7.70.3.3	<a href="#">nppiGetPerspectiveTransform</a>	821
7.70.3.4	<a href="#">nppiWarpPerspective_16u_AC4R</a>	821
7.70.3.5	<a href="#">nppiWarpPerspective_16u_C1R</a>	822
7.70.3.6	<a href="#">nppiWarpPerspective_16u_C3R</a>	822
7.70.3.7	<a href="#">nppiWarpPerspective_16u_C4R</a>	823
7.70.3.8	<a href="#">nppiWarpPerspective_16u_P3R</a>	823
7.70.3.9	<a href="#">nppiWarpPerspective_16u_P4R</a>	824
7.70.3.10	<a href="#">nppiWarpPerspective_32f_AC4R</a>	824
7.70.3.11	<a href="#">nppiWarpPerspective_32f_C1R</a>	825
7.70.3.12	<a href="#">nppiWarpPerspective_32f_C3R</a>	825
7.70.3.13	<a href="#">nppiWarpPerspective_32f_C4R</a>	826
7.70.3.14	<a href="#">nppiWarpPerspective_32f_P3R</a>	826
7.70.3.15	<a href="#">nppiWarpPerspective_32f_P4R</a>	827
7.70.3.16	<a href="#">nppiWarpPerspective_32s_AC4R</a>	827
7.70.3.17	<a href="#">nppiWarpPerspective_32s_C1R</a>	828
7.70.3.18	<a href="#">nppiWarpPerspective_32s_C3R</a>	828
7.70.3.19	<a href="#">nppiWarpPerspective_32s_C4R</a>	829
7.70.3.20	<a href="#">nppiWarpPerspective_32s_P3R</a>	829
7.70.3.21	<a href="#">nppiWarpPerspective_32s_P4R</a>	830
7.70.3.22	<a href="#">nppiWarpPerspective_8u_AC4R</a>	830
7.70.3.23	<a href="#">nppiWarpPerspective_8u_C1R</a>	831
7.70.3.24	<a href="#">nppiWarpPerspective_8u_C3R</a>	831
7.70.3.25	<a href="#">nppiWarpPerspective_8u_C4R</a>	832
7.70.3.26	<a href="#">nppiWarpPerspective_8u_P3R</a>	832
7.70.3.27	<a href="#">nppiWarpPerspective_8u_P4R</a>	833
7.70.3.28	<a href="#">nppiWarpPerspectiveBack_16u_AC4R</a>	833
7.70.3.29	<a href="#">nppiWarpPerspectiveBack_16u_C1R</a>	834
7.70.3.30	<a href="#">nppiWarpPerspectiveBack_16u_C3R</a>	834
7.70.3.31	<a href="#">nppiWarpPerspectiveBack_16u_C4R</a>	835
7.70.3.32	<a href="#">nppiWarpPerspectiveBack_16u_P3R</a>	835
7.70.3.33	<a href="#">nppiWarpPerspectiveBack_16u_P4R</a>	836
7.70.3.34	<a href="#">nppiWarpPerspectiveBack_32f_AC4R</a>	836
7.70.3.35	<a href="#">nppiWarpPerspectiveBack_32f_C1R</a>	837
7.70.3.36	<a href="#">nppiWarpPerspectiveBack_32f_C3R</a>	837
7.70.3.37	<a href="#">nppiWarpPerspectiveBack_32f_C4R</a>	838

7.70.3.38	<a href="#">nppiWarpPerspectiveBack_32f_P3R</a>	838
7.70.3.39	<a href="#">nppiWarpPerspectiveBack_32f_P4R</a>	839
7.70.3.40	<a href="#">nppiWarpPerspectiveBack_32s_AC4R</a>	839
7.70.3.41	<a href="#">nppiWarpPerspectiveBack_32s_C1R</a>	840
7.70.3.42	<a href="#">nppiWarpPerspectiveBack_32s_C3R</a>	840
7.70.3.43	<a href="#">nppiWarpPerspectiveBack_32s_C4R</a>	841
7.70.3.44	<a href="#">nppiWarpPerspectiveBack_32s_P3R</a>	841
7.70.3.45	<a href="#">nppiWarpPerspectiveBack_32s_P4R</a>	842
7.70.3.46	<a href="#">nppiWarpPerspectiveBack_8u_AC4R</a>	842
7.70.3.47	<a href="#">nppiWarpPerspectiveBack_8u_C1R</a>	843
7.70.3.48	<a href="#">nppiWarpPerspectiveBack_8u_C3R</a>	843
7.70.3.49	<a href="#">nppiWarpPerspectiveBack_8u_C4R</a>	844
7.70.3.50	<a href="#">nppiWarpPerspectiveBack_8u_P3R</a>	844
7.70.3.51	<a href="#">nppiWarpPerspectiveBack_8u_P4R</a>	845
7.70.3.52	<a href="#">nppiWarpPerspectiveQuad_16u_AC4R</a>	845
7.70.3.53	<a href="#">nppiWarpPerspectiveQuad_16u_C1R</a>	846
7.70.3.54	<a href="#">nppiWarpPerspectiveQuad_16u_C3R</a>	846
7.70.3.55	<a href="#">nppiWarpPerspectiveQuad_16u_C4R</a>	847
7.70.3.56	<a href="#">nppiWarpPerspectiveQuad_16u_P3R</a>	847
7.70.3.57	<a href="#">nppiWarpPerspectiveQuad_16u_P4R</a>	848
7.70.3.58	<a href="#">nppiWarpPerspectiveQuad_32f_AC4R</a>	848
7.70.3.59	<a href="#">nppiWarpPerspectiveQuad_32f_C1R</a>	849
7.70.3.60	<a href="#">nppiWarpPerspectiveQuad_32f_C3R</a>	849
7.70.3.61	<a href="#">nppiWarpPerspectiveQuad_32f_C4R</a>	850
7.70.3.62	<a href="#">nppiWarpPerspectiveQuad_32f_P3R</a>	850
7.70.3.63	<a href="#">nppiWarpPerspectiveQuad_32f_P4R</a>	851
7.70.3.64	<a href="#">nppiWarpPerspectiveQuad_32s_AC4R</a>	851
7.70.3.65	<a href="#">nppiWarpPerspectiveQuad_32s_C1R</a>	852
7.70.3.66	<a href="#">nppiWarpPerspectiveQuad_32s_C3R</a>	852
7.70.3.67	<a href="#">nppiWarpPerspectiveQuad_32s_C4R</a>	853
7.70.3.68	<a href="#">nppiWarpPerspectiveQuad_32s_P3R</a>	853
7.70.3.69	<a href="#">nppiWarpPerspectiveQuad_32s_P4R</a>	854
7.70.3.70	<a href="#">nppiWarpPerspectiveQuad_8u_AC4R</a>	854
7.70.3.71	<a href="#">nppiWarpPerspectiveQuad_8u_C1R</a>	855
7.70.3.72	<a href="#">nppiWarpPerspectiveQuad_8u_C3R</a>	855
7.70.3.73	<a href="#">nppiWarpPerspectiveQuad_8u_C4R</a>	856

---

7.70.3.74	<code>nppiWarpPerspectiveQuad_8u_P3R</code>	856
7.70.3.75	<code>nppiWarpPerspectiveQuad_8u_P4R</code>	857
7.71	Linear Transforms	858
7.71.1	Detailed Description	858
7.72	Fourier Transforms	859
7.72.1	Function Documentation	859
7.72.1.1	<code>nppiMagnitude_32fc32f_C1R</code>	859
7.72.1.2	<code>nppiMagnitudeSqr_32fc32f_C1R</code>	859
7.73	Morphological Operations	861
7.73.1	Detailed Description	861
7.74	Dilation And Erosion	862
7.74.1	Function Documentation	862
7.74.1.1	<code>nppiDilate_8u_C1R</code>	862
7.74.1.2	<code>nppiDilate_8u_C4R</code>	863
7.74.1.3	<code>nppiErode_8u_C1R</code>	863
7.74.1.4	<code>nppiErode_8u_C4R</code>	864
7.75	Statistics Functions	865
7.75.1	Detailed Description	865
7.76	Sum	866
7.76.1	Function Documentation	868
7.76.1.1	<code>nppiSum_16s_AC4R</code>	868
7.76.1.2	<code>nppiSum_16s_C1R</code>	869
7.76.1.3	<code>nppiSum_16s_C3R</code>	869
7.76.1.4	<code>nppiSum_16s_C4R</code>	869
7.76.1.5	<code>nppiSum_16u_AC4R</code>	870
7.76.1.6	<code>nppiSum_16u_C1R</code>	870
7.76.1.7	<code>nppiSum_16u_C3R</code>	870
7.76.1.8	<code>nppiSum_16u_C4R</code>	871
7.76.1.9	<code>nppiSum_32f_AC4R</code>	871
7.76.1.10	<code>nppiSum_32f_C1R</code>	871
7.76.1.11	<code>nppiSum_32f_C3R</code>	872
7.76.1.12	<code>nppiSum_32f_C4R</code>	872
7.76.1.13	<code>nppiSum_8u64s_C1R</code>	872
7.76.1.14	<code>nppiSum_8u64s_C4R</code>	873
7.76.1.15	<code>nppiSum_8u_AC4R</code>	873
7.76.1.16	<code>nppiSum_8u_C1R</code>	873

7.76.1.17	nppiSum_8u_C3R	874
7.76.1.18	nppiSum_8u_C4R	874
7.76.1.19	nppiSumGetBufferHostSize_16s_AC4R	874
7.76.1.20	nppiSumGetBufferHostSize_16s_C1R	875
7.76.1.21	nppiSumGetBufferHostSize_16s_C3R	875
7.76.1.22	nppiSumGetBufferHostSize_16s_C4R	875
7.76.1.23	nppiSumGetBufferHostSize_16u_AC4R	875
7.76.1.24	nppiSumGetBufferHostSize_16u_C1R	876
7.76.1.25	nppiSumGetBufferHostSize_16u_C3R	876
7.76.1.26	nppiSumGetBufferHostSize_16u_C4R	876
7.76.1.27	nppiSumGetBufferHostSize_32f_AC4R	877
7.76.1.28	nppiSumGetBufferHostSize_32f_C1R	877
7.76.1.29	nppiSumGetBufferHostSize_32f_C3R	877
7.76.1.30	nppiSumGetBufferHostSize_32f_C4R	877
7.76.1.31	nppiSumGetBufferHostSize_8u64s_C1R	878
7.76.1.32	nppiSumGetBufferHostSize_8u64s_C4R	878
7.76.1.33	nppiSumGetBufferHostSize_8u_AC4R	878
7.76.1.34	nppiSumGetBufferHostSize_8u_C1R	879
7.76.1.35	nppiSumGetBufferHostSize_8u_C3R	879
7.76.1.36	nppiSumGetBufferHostSize_8u_C4R	879
7.77	Minimum	880
7.77.1	Function Documentation	884
7.77.1.1	nppiMin_16s_AC4R	884
7.77.1.2	nppiMin_16s_C1R	885
7.77.1.3	nppiMin_16s_C3R	885
7.77.1.4	nppiMin_16s_C4R	886
7.77.1.5	nppiMin_16u_AC4R	886
7.77.1.6	nppiMin_16u_C1R	886
7.77.1.7	nppiMin_16u_C3R	887
7.77.1.8	nppiMin_16u_C4R	887
7.77.1.9	nppiMin_32f_AC4R	887
7.77.1.10	nppiMin_32f_C1R	888
7.77.1.11	nppiMin_32f_C3R	888
7.77.1.12	nppiMin_32f_C4R	888
7.77.1.13	nppiMin_8u_AC4R	889
7.77.1.14	nppiMin_8u_C1R	889

---

7.77.1.15	<a href="#">nppiMin_8u_C3R</a>	889
7.77.1.16	<a href="#">nppiMin_8u_C4R</a>	890
7.77.1.17	<a href="#">nppiMinGetBufferHostSize_16s_AC4R</a>	890
7.77.1.18	<a href="#">nppiMinGetBufferHostSize_16s_C1R</a>	890
7.77.1.19	<a href="#">nppiMinGetBufferHostSize_16s_C3R</a>	891
7.77.1.20	<a href="#">nppiMinGetBufferHostSize_16s_C4R</a>	891
7.77.1.21	<a href="#">nppiMinGetBufferHostSize_16u_AC4R</a>	891
7.77.1.22	<a href="#">nppiMinGetBufferHostSize_16u_C1R</a>	891
7.77.1.23	<a href="#">nppiMinGetBufferHostSize_16u_C3R</a>	892
7.77.1.24	<a href="#">nppiMinGetBufferHostSize_16u_C4R</a>	892
7.77.1.25	<a href="#">nppiMinGetBufferHostSize_32f_AC4R</a>	892
7.77.1.26	<a href="#">nppiMinGetBufferHostSize_32f_C1R</a>	893
7.77.1.27	<a href="#">nppiMinGetBufferHostSize_32f_C3R</a>	893
7.77.1.28	<a href="#">nppiMinGetBufferHostSize_32f_C4R</a>	893
7.77.1.29	<a href="#">nppiMinGetBufferHostSize_8u_AC4R</a>	893
7.77.1.30	<a href="#">nppiMinGetBufferHostSize_8u_C1R</a>	894
7.77.1.31	<a href="#">nppiMinGetBufferHostSize_8u_C3R</a>	894
7.77.1.32	<a href="#">nppiMinGetBufferHostSize_8u_C4R</a>	894
7.77.1.33	<a href="#">nppiMinIndx_16s_AC4R</a>	895
7.77.1.34	<a href="#">nppiMinIndx_16s_C1R</a>	895
7.77.1.35	<a href="#">nppiMinIndx_16s_C3R</a>	895
7.77.1.36	<a href="#">nppiMinIndx_16s_C4R</a>	896
7.77.1.37	<a href="#">nppiMinIndx_16u_AC4R</a>	896
7.77.1.38	<a href="#">nppiMinIndx_16u_C1R</a>	897
7.77.1.39	<a href="#">nppiMinIndx_16u_C3R</a>	897
7.77.1.40	<a href="#">nppiMinIndx_16u_C4R</a>	898
7.77.1.41	<a href="#">nppiMinIndx_32f_AC4R</a>	898
7.77.1.42	<a href="#">nppiMinIndx_32f_C1R</a>	899
7.77.1.43	<a href="#">nppiMinIndx_32f_C3R</a>	899
7.77.1.44	<a href="#">nppiMinIndx_32f_C4R</a>	899
7.77.1.45	<a href="#">nppiMinIndx_8u_AC4R</a>	900
7.77.1.46	<a href="#">nppiMinIndx_8u_C1R</a>	900
7.77.1.47	<a href="#">nppiMinIndx_8u_C3R</a>	901
7.77.1.48	<a href="#">nppiMinIndx_8u_C4R</a>	901
7.77.1.49	<a href="#">nppiMinIndxGetBufferHostSize_16s_AC4R</a>	902
7.77.1.50	<a href="#">nppiMinIndxGetBufferHostSize_16s_C1R</a>	902

7.77.1.51	nppiMinIndxGetBufferHostSize_16s_C3R	902
7.77.1.52	nppiMinIndxGetBufferHostSize_16s_C4R	902
7.77.1.53	nppiMinIndxGetBufferHostSize_16u_AC4R	903
7.77.1.54	nppiMinIndxGetBufferHostSize_16u_C1R	903
7.77.1.55	nppiMinIndxGetBufferHostSize_16u_C3R	903
7.77.1.56	nppiMinIndxGetBufferHostSize_16u_C4R	904
7.77.1.57	nppiMinIndxGetBufferHostSize_32f_AC4R	904
7.77.1.58	nppiMinIndxGetBufferHostSize_32f_C1R	904
7.77.1.59	nppiMinIndxGetBufferHostSize_32f_C3R	904
7.77.1.60	nppiMinIndxGetBufferHostSize_32f_C4R	905
7.77.1.61	nppiMinIndxGetBufferHostSize_8u_AC4R	905
7.77.1.62	nppiMinIndxGetBufferHostSize_8u_C1R	905
7.77.1.63	nppiMinIndxGetBufferHostSize_8u_C3R	906
7.77.1.64	nppiMinIndxGetBufferHostSize_8u_C4R	906
7.78	Maximum	907
7.78.1	Function Documentation	911
7.78.1.1	nppiMax_16s_AC4R	911
7.78.1.2	nppiMax_16s_C1R	912
7.78.1.3	nppiMax_16s_C3R	912
7.78.1.4	nppiMax_16s_C4R	913
7.78.1.5	nppiMax_16u_AC4R	913
7.78.1.6	nppiMax_16u_C1R	913
7.78.1.7	nppiMax_16u_C3R	914
7.78.1.8	nppiMax_16u_C4R	914
7.78.1.9	nppiMax_32f_AC4R	914
7.78.1.10	nppiMax_32f_C1R	915
7.78.1.11	nppiMax_32f_C3R	915
7.78.1.12	nppiMax_32f_C4R	915
7.78.1.13	nppiMax_8u_AC4R	916
7.78.1.14	nppiMax_8u_C1R	916
7.78.1.15	nppiMax_8u_C3R	916
7.78.1.16	nppiMax_8u_C4R	917
7.78.1.17	nppiMaxGetBufferHostSize_16s_AC4R	917
7.78.1.18	nppiMaxGetBufferHostSize_16s_C1R	917
7.78.1.19	nppiMaxGetBufferHostSize_16s_C3R	918
7.78.1.20	nppiMaxGetBufferHostSize_16s_C4R	918

7.78.1.21	nppiMaxGetBufferHostSize_16u_AC4R	918
7.78.1.22	nppiMaxGetBufferHostSize_16u_C1R	918
7.78.1.23	nppiMaxGetBufferHostSize_16u_C3R	919
7.78.1.24	nppiMaxGetBufferHostSize_16u_C4R	919
7.78.1.25	nppiMaxGetBufferHostSize_32f_AC4R	919
7.78.1.26	nppiMaxGetBufferHostSize_32f_C1R	920
7.78.1.27	nppiMaxGetBufferHostSize_32f_C3R	920
7.78.1.28	nppiMaxGetBufferHostSize_32f_C4R	920
7.78.1.29	nppiMaxGetBufferHostSize_8u_AC4R	920
7.78.1.30	nppiMaxGetBufferHostSize_8u_C1R	921
7.78.1.31	nppiMaxGetBufferHostSize_8u_C3R	921
7.78.1.32	nppiMaxGetBufferHostSize_8u_C4R	921
7.78.1.33	nppiMaxIndx_16s_AC4R	922
7.78.1.34	nppiMaxIndx_16s_C1R	922
7.78.1.35	nppiMaxIndx_16s_C3R	922
7.78.1.36	nppiMaxIndx_16s_C4R	923
7.78.1.37	nppiMaxIndx_16u_AC4R	923
7.78.1.38	nppiMaxIndx_16u_C1R	924
7.78.1.39	nppiMaxIndx_16u_C3R	924
7.78.1.40	nppiMaxIndx_16u_C4R	925
7.78.1.41	nppiMaxIndx_32f_AC4R	925
7.78.1.42	nppiMaxIndx_32f_C1R	926
7.78.1.43	nppiMaxIndx_32f_C3R	926
7.78.1.44	nppiMaxIndx_32f_C4R	926
7.78.1.45	nppiMaxIndx_8u_AC4R	927
7.78.1.46	nppiMaxIndx_8u_C1R	927
7.78.1.47	nppiMaxIndx_8u_C3R	928
7.78.1.48	nppiMaxIndx_8u_C4R	928
7.78.1.49	nppiMaxIndxGetBufferHostSize_16s_AC4R	929
7.78.1.50	nppiMaxIndxGetBufferHostSize_16s_C1R	929
7.78.1.51	nppiMaxIndxGetBufferHostSize_16s_C3R	929
7.78.1.52	nppiMaxIndxGetBufferHostSize_16s_C4R	929
7.78.1.53	nppiMaxIndxGetBufferHostSize_16u_AC4R	930
7.78.1.54	nppiMaxIndxGetBufferHostSize_16u_C1R	930
7.78.1.55	nppiMaxIndxGetBufferHostSize_16u_C3R	930
7.78.1.56	nppiMaxIndxGetBufferHostSize_16u_C4R	931

---

7.78.1.57	<a href="#">nppiMaxIdxGetBufferHostSize_32f_AC4R</a>	931
7.78.1.58	<a href="#">nppiMaxIdxGetBufferHostSize_32f_C1R</a>	931
7.78.1.59	<a href="#">nppiMaxIdxGetBufferHostSize_32f_C3R</a>	931
7.78.1.60	<a href="#">nppiMaxIdxGetBufferHostSize_32f_C4R</a>	932
7.78.1.61	<a href="#">nppiMaxIdxGetBufferHostSize_8u_AC4R</a>	932
7.78.1.62	<a href="#">nppiMaxIdxGetBufferHostSize_8u_C1R</a>	932
7.78.1.63	<a href="#">nppiMaxIdxGetBufferHostSize_8u_C3R</a>	933
7.78.1.64	<a href="#">nppiMaxIdxGetBufferHostSize_8u_C4R</a>	933
7.79	<a href="#">Minimum_Maximum</a>	934
7.79.1	<a href="#">Function Documentation</a>	939
7.79.1.1	<a href="#">nppiMinMax_16s_AC4R</a>	939
7.79.1.2	<a href="#">nppiMinMax_16s_C1R</a>	940
7.79.1.3	<a href="#">nppiMinMax_16s_C3R</a>	940
7.79.1.4	<a href="#">nppiMinMax_16s_C4R</a>	940
7.79.1.5	<a href="#">nppiMinMax_16u_AC4R</a>	941
7.79.1.6	<a href="#">nppiMinMax_16u_C1R</a>	941
7.79.1.7	<a href="#">nppiMinMax_16u_C3R</a>	942
7.79.1.8	<a href="#">nppiMinMax_16u_C4R</a>	942
7.79.1.9	<a href="#">nppiMinMax_32f_AC4R</a>	942
7.79.1.10	<a href="#">nppiMinMax_32f_C1R</a>	943
7.79.1.11	<a href="#">nppiMinMax_32f_C3R</a>	943
7.79.1.12	<a href="#">nppiMinMax_32f_C4R</a>	944
7.79.1.13	<a href="#">nppiMinMax_8u_AC4R</a>	944
7.79.1.14	<a href="#">nppiMinMax_8u_C1R</a>	944
7.79.1.15	<a href="#">nppiMinMax_8u_C3R</a>	945
7.79.1.16	<a href="#">nppiMinMax_8u_C4R</a>	945
7.79.1.17	<a href="#">nppiMinMaxGetBufferHostSize_16s_AC4R</a>	946
7.79.1.18	<a href="#">nppiMinMaxGetBufferHostSize_16s_C1R</a>	946
7.79.1.19	<a href="#">nppiMinMaxGetBufferHostSize_16s_C3R</a>	946
7.79.1.20	<a href="#">nppiMinMaxGetBufferHostSize_16s_C4R</a>	946
7.79.1.21	<a href="#">nppiMinMaxGetBufferHostSize_16u_AC4R</a>	947
7.79.1.22	<a href="#">nppiMinMaxGetBufferHostSize_16u_C1R</a>	947
7.79.1.23	<a href="#">nppiMinMaxGetBufferHostSize_16u_C3R</a>	947
7.79.1.24	<a href="#">nppiMinMaxGetBufferHostSize_16u_C4R</a>	948
7.79.1.25	<a href="#">nppiMinMaxGetBufferHostSize_32f_AC4R</a>	948
7.79.1.26	<a href="#">nppiMinMaxGetBufferHostSize_32f_C1R</a>	948

---

---

7.79.1.27	<a href="#">nppiMinMaxGetBufferHostSize_32f_C3R</a>	948
7.79.1.28	<a href="#">nppiMinMaxGetBufferHostSize_32f_C4R</a>	949
7.79.1.29	<a href="#">nppiMinMaxGetBufferHostSize_8u_AC4R</a>	949
7.79.1.30	<a href="#">nppiMinMaxGetBufferHostSize_8u_C1R</a>	949
7.79.1.31	<a href="#">nppiMinMaxGetBufferHostSize_8u_C3R</a>	950
7.79.1.32	<a href="#">nppiMinMaxGetBufferHostSize_8u_C4R</a>	950
7.79.1.33	<a href="#">nppiMinMaxIndx_16u_C1MR</a>	950
7.79.1.34	<a href="#">nppiMinMaxIndx_16u_C1R</a>	951
7.79.1.35	<a href="#">nppiMinMaxIndx_16u_C3CMR</a>	951
7.79.1.36	<a href="#">nppiMinMaxIndx_16u_C3CR</a>	952
7.79.1.37	<a href="#">nppiMinMaxIndx_32f_C1MR</a>	953
7.79.1.38	<a href="#">nppiMinMaxIndx_32f_C1R</a>	953
7.79.1.39	<a href="#">nppiMinMaxIndx_32f_C3CMR</a>	954
7.79.1.40	<a href="#">nppiMinMaxIndx_32f_C3CR</a>	954
7.79.1.41	<a href="#">nppiMinMaxIndx_8s_C1MR</a>	955
7.79.1.42	<a href="#">nppiMinMaxIndx_8s_C1R</a>	956
7.79.1.43	<a href="#">nppiMinMaxIndx_8s_C3CMR</a>	956
7.79.1.44	<a href="#">nppiMinMaxIndx_8s_C3CR</a>	957
7.79.1.45	<a href="#">nppiMinMaxIndx_8u_C1MR</a>	957
7.79.1.46	<a href="#">nppiMinMaxIndx_8u_C1R</a>	958
7.79.1.47	<a href="#">nppiMinMaxIndx_8u_C3CMR</a>	958
7.79.1.48	<a href="#">nppiMinMaxIndx_8u_C3CR</a>	959
7.79.1.49	<a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C1MR</a>	960
7.79.1.50	<a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C1R</a>	960
7.79.1.51	<a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C3CMR</a>	960
7.79.1.52	<a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C3CR</a>	960
7.79.1.53	<a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C1MR</a>	961
7.79.1.54	<a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C1R</a>	961
7.79.1.55	<a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C3CMR</a>	961
7.79.1.56	<a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C3CR</a>	962
7.79.1.57	<a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C1MR</a>	962
7.79.1.58	<a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C1R</a>	962
7.79.1.59	<a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C3CMR</a>	962
7.79.1.60	<a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C3CR</a>	963
7.79.1.61	<a href="#">nppiMinMaxIndxGetBufferHostSize_8u_C1MR</a>	963
7.79.1.62	<a href="#">nppiMinMaxIndxGetBufferHostSize_8u_C1R</a>	963

7.79.1.63	<a href="#">nppiMinMaxIndxGetBufferHostSize_8u_C3CMR</a>	964
7.79.1.64	<a href="#">nppiMinMaxIndxGetBufferHostSize_8u_C3CR</a>	964
7.80	<a href="#">Mean</a>	965
7.80.1	<a href="#">Function Documentation</a>	968
7.80.1.1	<a href="#">nppiMean_16s_AC4R</a>	968
7.80.1.2	<a href="#">nppiMean_16s_C1R</a>	969
7.80.1.3	<a href="#">nppiMean_16s_C3R</a>	969
7.80.1.4	<a href="#">nppiMean_16s_C4R</a>	969
7.80.1.5	<a href="#">nppiMean_16u_AC4R</a>	970
7.80.1.6	<a href="#">nppiMean_16u_C1MR</a>	970
7.80.1.7	<a href="#">nppiMean_16u_C1R</a>	971
7.80.1.8	<a href="#">nppiMean_16u_C3CMR</a>	971
7.80.1.9	<a href="#">nppiMean_16u_C3R</a>	971
7.80.1.10	<a href="#">nppiMean_16u_C4R</a>	972
7.80.1.11	<a href="#">nppiMean_32f_AC4R</a>	972
7.80.1.12	<a href="#">nppiMean_32f_C1MR</a>	973
7.80.1.13	<a href="#">nppiMean_32f_C1R</a>	973
7.80.1.14	<a href="#">nppiMean_32f_C3CMR</a>	973
7.80.1.15	<a href="#">nppiMean_32f_C3R</a>	974
7.80.1.16	<a href="#">nppiMean_32f_C4R</a>	974
7.80.1.17	<a href="#">nppiMean_8s_C1MR</a>	975
7.80.1.18	<a href="#">nppiMean_8s_C3CMR</a>	975
7.80.1.19	<a href="#">nppiMean_8u_AC4R</a>	976
7.80.1.20	<a href="#">nppiMean_8u_C1MR</a>	976
7.80.1.21	<a href="#">nppiMean_8u_C1R</a>	976
7.80.1.22	<a href="#">nppiMean_8u_C3CMR</a>	977
7.80.1.23	<a href="#">nppiMean_8u_C3R</a>	977
7.80.1.24	<a href="#">nppiMean_8u_C4R</a>	978
7.80.1.25	<a href="#">nppiMeanGetBufferHostSize_16s_AC4R</a>	978
7.80.1.26	<a href="#">nppiMeanGetBufferHostSize_16s_C1R</a>	978
7.80.1.27	<a href="#">nppiMeanGetBufferHostSize_16s_C3R</a>	979
7.80.1.28	<a href="#">nppiMeanGetBufferHostSize_16s_C4R</a>	979
7.80.1.29	<a href="#">nppiMeanGetBufferHostSize_16u_AC4R</a>	979
7.80.1.30	<a href="#">nppiMeanGetBufferHostSize_16u_C1MR</a>	979
7.80.1.31	<a href="#">nppiMeanGetBufferHostSize_16u_C1R</a>	980
7.80.1.32	<a href="#">nppiMeanGetBufferHostSize_16u_C3CMR</a>	980

7.80.1.33	<code>nppiMeanGetBufferHostSize_16u_C3R</code>	980
7.80.1.34	<code>nppiMeanGetBufferHostSize_16u_C4R</code>	981
7.80.1.35	<code>nppiMeanGetBufferHostSize_32f_AC4R</code>	981
7.80.1.36	<code>nppiMeanGetBufferHostSize_32f_C1MR</code>	981
7.80.1.37	<code>nppiMeanGetBufferHostSize_32f_C1R</code>	981
7.80.1.38	<code>nppiMeanGetBufferHostSize_32f_C3CMR</code>	982
7.80.1.39	<code>nppiMeanGetBufferHostSize_32f_C3R</code>	982
7.80.1.40	<code>nppiMeanGetBufferHostSize_32f_C4R</code>	982
7.80.1.41	<code>nppiMeanGetBufferHostSize_8s_C1MR</code>	983
7.80.1.42	<code>nppiMeanGetBufferHostSize_8s_C3CMR</code>	983
7.80.1.43	<code>nppiMeanGetBufferHostSize_8u_AC4R</code>	983
7.80.1.44	<code>nppiMeanGetBufferHostSize_8u_C1MR</code>	983
7.80.1.45	<code>nppiMeanGetBufferHostSize_8u_C1R</code>	984
7.80.1.46	<code>nppiMeanGetBufferHostSize_8u_C3CMR</code>	984
7.80.1.47	<code>nppiMeanGetBufferHostSize_8u_C3R</code>	984
7.80.1.48	<code>nppiMeanGetBufferHostSize_8u_C4R</code>	985
7.81	Mean And Standard Deviation	986
7.81.1	Function Documentation	988
7.81.1.1	<code>nppiMean_StdDev_16u_C1MR</code>	988
7.81.1.2	<code>nppiMean_StdDev_16u_C1R</code>	989
7.81.1.3	<code>nppiMean_StdDev_16u_C3CMR</code>	989
7.81.1.4	<code>nppiMean_StdDev_16u_C3CR</code>	990
7.81.1.5	<code>nppiMean_StdDev_32f_C1MR</code>	990
7.81.1.6	<code>nppiMean_StdDev_32f_C1R</code>	991
7.81.1.7	<code>nppiMean_StdDev_32f_C3CMR</code>	991
7.81.1.8	<code>nppiMean_StdDev_32f_C3CR</code>	992
7.81.1.9	<code>nppiMean_StdDev_8s_C1MR</code>	992
7.81.1.10	<code>nppiMean_StdDev_8s_C1R</code>	993
7.81.1.11	<code>nppiMean_StdDev_8s_C3CMR</code>	993
7.81.1.12	<code>nppiMean_StdDev_8s_C3CR</code>	994
7.81.1.13	<code>nppiMean_StdDev_8u_C1MR</code>	994
7.81.1.14	<code>nppiMean_StdDev_8u_C1R</code>	995
7.81.1.15	<code>nppiMean_StdDev_8u_C3CMR</code>	995
7.81.1.16	<code>nppiMean_StdDev_8u_C3CR</code>	996
7.81.1.17	<code>nppiMeanStdDevGetBufferHostSize_16u_C1MR</code>	996
7.81.1.18	<code>nppiMeanStdDevGetBufferHostSize_16u_C1R</code>	996

7.81.1.19	nppiMeanStdDevGetBufferHostSize_16u_C3CMR	997
7.81.1.20	nppiMeanStdDevGetBufferHostSize_16u_C3CR	997
7.81.1.21	nppiMeanStdDevGetBufferHostSize_32f_C1MR	997
7.81.1.22	nppiMeanStdDevGetBufferHostSize_32f_C1R	997
7.81.1.23	nppiMeanStdDevGetBufferHostSize_32f_C3CMR	998
7.81.1.24	nppiMeanStdDevGetBufferHostSize_32f_C3CR	998
7.81.1.25	nppiMeanStdDevGetBufferHostSize_8s_C1MR	998
7.81.1.26	nppiMeanStdDevGetBufferHostSize_8s_C1R	999
7.81.1.27	nppiMeanStdDevGetBufferHostSize_8s_C3CMR	999
7.81.1.28	nppiMeanStdDevGetBufferHostSize_8s_C3CR	999
7.81.1.29	nppiMeanStdDevGetBufferHostSize_8u_C1MR	999
7.81.1.30	nppiMeanStdDevGetBufferHostSize_8u_C1R	1000
7.81.1.31	nppiMeanStdDevGetBufferHostSize_8u_C3CMR	1000
7.81.1.32	nppiMeanStdDevGetBufferHostSize_8u_C3CR	1000
7.82	Infinity Norm	1001
7.82.1	Function Documentation	1004
7.82.1.1	nppiNorm_Inf_16s_AC4R	1004
7.82.1.2	nppiNorm_Inf_16s_C1R	1005
7.82.1.3	nppiNorm_Inf_16s_C3R	1005
7.82.1.4	nppiNorm_Inf_16s_C4R	1005
7.82.1.5	nppiNorm_Inf_16u_AC4R	1006
7.82.1.6	nppiNorm_Inf_16u_C1MR	1006
7.82.1.7	nppiNorm_Inf_16u_C1R	1007
7.82.1.8	nppiNorm_Inf_16u_C3CMR	1007
7.82.1.9	nppiNorm_Inf_16u_C3R	1007
7.82.1.10	nppiNorm_Inf_16u_C4R	1008
7.82.1.11	nppiNorm_Inf_32f_AC4R	1008
7.82.1.12	nppiNorm_Inf_32f_C1MR	1009
7.82.1.13	nppiNorm_Inf_32f_C1R	1009
7.82.1.14	nppiNorm_Inf_32f_C3CMR	1009
7.82.1.15	nppiNorm_Inf_32f_C3R	1010
7.82.1.16	nppiNorm_Inf_32f_C4R	1010
7.82.1.17	nppiNorm_Inf_32s_C1R	1011
7.82.1.18	nppiNorm_Inf_8s_C1MR	1011
7.82.1.19	nppiNorm_Inf_8s_C3CMR	1011
7.82.1.20	nppiNorm_Inf_8u_AC4R	1012

7.82.1.21	nppiNorm_Inf_8u_C1MR	1012
7.82.1.22	nppiNorm_Inf_8u_C1R	1013
7.82.1.23	nppiNorm_Inf_8u_C3CMR	1013
7.82.1.24	nppiNorm_Inf_8u_C3R	1013
7.82.1.25	nppiNorm_Inf_8u_C4R	1014
7.82.1.26	nppiNormInfGetBufferHostSize_16s_AC4R	1014
7.82.1.27	nppiNormInfGetBufferHostSize_16s_C1R	1014
7.82.1.28	nppiNormInfGetBufferHostSize_16s_C3R	1015
7.82.1.29	nppiNormInfGetBufferHostSize_16s_C4R	1015
7.82.1.30	nppiNormInfGetBufferHostSize_16u_AC4R	1015
7.82.1.31	nppiNormInfGetBufferHostSize_16u_C1MR	1016
7.82.1.32	nppiNormInfGetBufferHostSize_16u_C1R	1016
7.82.1.33	nppiNormInfGetBufferHostSize_16u_C3CMR	1016
7.82.1.34	nppiNormInfGetBufferHostSize_16u_C3R	1016
7.82.1.35	nppiNormInfGetBufferHostSize_16u_C4R	1017
7.82.1.36	nppiNormInfGetBufferHostSize_32f_AC4R	1017
7.82.1.37	nppiNormInfGetBufferHostSize_32f_C1MR	1017
7.82.1.38	nppiNormInfGetBufferHostSize_32f_C1R	1018
7.82.1.39	nppiNormInfGetBufferHostSize_32f_C3CMR	1018
7.82.1.40	nppiNormInfGetBufferHostSize_32f_C3R	1018
7.82.1.41	nppiNormInfGetBufferHostSize_32f_C4R	1018
7.82.1.42	nppiNormInfGetBufferHostSize_32s_C1R	1019
7.82.1.43	nppiNormInfGetBufferHostSize_8s_C1MR	1019
7.82.1.44	nppiNormInfGetBufferHostSize_8s_C3CMR	1019
7.82.1.45	nppiNormInfGetBufferHostSize_8u_AC4R	1020
7.82.1.46	nppiNormInfGetBufferHostSize_8u_C1MR	1020
7.82.1.47	nppiNormInfGetBufferHostSize_8u_C1R	1020
7.82.1.48	nppiNormInfGetBufferHostSize_8u_C3CMR	1020
7.82.1.49	nppiNormInfGetBufferHostSize_8u_C3R	1021
7.82.1.50	nppiNormInfGetBufferHostSize_8u_C4R	1021
7.83	L1 Norm	1022
7.83.1	Function Documentation	1025
7.83.1.1	nppiNorm_L1_16s_AC4R	1025
7.83.1.2	nppiNorm_L1_16s_C1R	1026
7.83.1.3	nppiNorm_L1_16s_C3R	1026
7.83.1.4	nppiNorm_L1_16s_C4R	1026

7.83.1.5	nppiNorm_L1_16u_AC4R	1027
7.83.1.6	nppiNorm_L1_16u_C1MR	1027
7.83.1.7	nppiNorm_L1_16u_C1R	1027
7.83.1.8	nppiNorm_L1_16u_C3CMR	1028
7.83.1.9	nppiNorm_L1_16u_C3R	1028
7.83.1.10	nppiNorm_L1_16u_C4R	1029
7.83.1.11	nppiNorm_L1_32f_AC4R	1029
7.83.1.12	nppiNorm_L1_32f_C1MR	1029
7.83.1.13	nppiNorm_L1_32f_C1R	1030
7.83.1.14	nppiNorm_L1_32f_C3CMR	1030
7.83.1.15	nppiNorm_L1_32f_C3R	1031
7.83.1.16	nppiNorm_L1_32f_C4R	1031
7.83.1.17	nppiNorm_L1_8s_C1MR	1031
7.83.1.18	nppiNorm_L1_8s_C3CMR	1032
7.83.1.19	nppiNorm_L1_8u_AC4R	1032
7.83.1.20	nppiNorm_L1_8u_C1MR	1033
7.83.1.21	nppiNorm_L1_8u_C1R	1033
7.83.1.22	nppiNorm_L1_8u_C3CMR	1033
7.83.1.23	nppiNorm_L1_8u_C3R	1034
7.83.1.24	nppiNorm_L1_8u_C4R	1034
7.83.1.25	nppiNormL1GetBufferHostSize_16s_AC4R	1035
7.83.1.26	nppiNormL1GetBufferHostSize_16s_C1R	1035
7.83.1.27	nppiNormL1GetBufferHostSize_16s_C3R	1035
7.83.1.28	nppiNormL1GetBufferHostSize_16s_C4R	1035
7.83.1.29	nppiNormL1GetBufferHostSize_16u_AC4R	1036
7.83.1.30	nppiNormL1GetBufferHostSize_16u_C1MR	1036
7.83.1.31	nppiNormL1GetBufferHostSize_16u_C1R	1036
7.83.1.32	nppiNormL1GetBufferHostSize_16u_C3CMR	1037
7.83.1.33	nppiNormL1GetBufferHostSize_16u_C3R	1037
7.83.1.34	nppiNormL1GetBufferHostSize_16u_C4R	1037
7.83.1.35	nppiNormL1GetBufferHostSize_32f_AC4R	1037
7.83.1.36	nppiNormL1GetBufferHostSize_32f_C1MR	1038
7.83.1.37	nppiNormL1GetBufferHostSize_32f_C1R	1038
7.83.1.38	nppiNormL1GetBufferHostSize_32f_C3CMR	1038
7.83.1.39	nppiNormL1GetBufferHostSize_32f_C3R	1039
7.83.1.40	nppiNormL1GetBufferHostSize_32f_C4R	1039

---

7.83.1.41	<a href="#">nppiNormL1GetBufferHostSize_8s_C1MR</a>	1039
7.83.1.42	<a href="#">nppiNormL1GetBufferHostSize_8s_C3CMR</a>	1039
7.83.1.43	<a href="#">nppiNormL1GetBufferHostSize_8u_AC4R</a>	1040
7.83.1.44	<a href="#">nppiNormL1GetBufferHostSize_8u_C1MR</a>	1040
7.83.1.45	<a href="#">nppiNormL1GetBufferHostSize_8u_C1R</a>	1040
7.83.1.46	<a href="#">nppiNormL1GetBufferHostSize_8u_C3CMR</a>	1041
7.83.1.47	<a href="#">nppiNormL1GetBufferHostSize_8u_C3R</a>	1041
7.83.1.48	<a href="#">nppiNormL1GetBufferHostSize_8u_C4R</a>	1041
7.84	<a href="#">L2 Norm</a>	1042
7.84.1	<a href="#">Function Documentation</a>	1045
7.84.1.1	<a href="#">nppiNorm_L2_16s_AC4R</a>	1045
7.84.1.2	<a href="#">nppiNorm_L2_16s_C1R</a>	1046
7.84.1.3	<a href="#">nppiNorm_L2_16s_C3R</a>	1046
7.84.1.4	<a href="#">nppiNorm_L2_16s_C4R</a>	1046
7.84.1.5	<a href="#">nppiNorm_L2_16u_AC4R</a>	1047
7.84.1.6	<a href="#">nppiNorm_L2_16u_C1MR</a>	1047
7.84.1.7	<a href="#">nppiNorm_L2_16u_C1R</a>	1047
7.84.1.8	<a href="#">nppiNorm_L2_16u_C3CMR</a>	1048
7.84.1.9	<a href="#">nppiNorm_L2_16u_C3R</a>	1048
7.84.1.10	<a href="#">nppiNorm_L2_16u_C4R</a>	1049
7.84.1.11	<a href="#">nppiNorm_L2_32f_AC4R</a>	1049
7.84.1.12	<a href="#">nppiNorm_L2_32f_C1MR</a>	1049
7.84.1.13	<a href="#">nppiNorm_L2_32f_C1R</a>	1050
7.84.1.14	<a href="#">nppiNorm_L2_32f_C3CMR</a>	1050
7.84.1.15	<a href="#">nppiNorm_L2_32f_C3R</a>	1051
7.84.1.16	<a href="#">nppiNorm_L2_32f_C4R</a>	1051
7.84.1.17	<a href="#">nppiNorm_L2_8s_C1MR</a>	1051
7.84.1.18	<a href="#">nppiNorm_L2_8s_C3CMR</a>	1052
7.84.1.19	<a href="#">nppiNorm_L2_8u_AC4R</a>	1052
7.84.1.20	<a href="#">nppiNorm_L2_8u_C1MR</a>	1053
7.84.1.21	<a href="#">nppiNorm_L2_8u_C1R</a>	1053
7.84.1.22	<a href="#">nppiNorm_L2_8u_C3CMR</a>	1053
7.84.1.23	<a href="#">nppiNorm_L2_8u_C3R</a>	1054
7.84.1.24	<a href="#">nppiNorm_L2_8u_C4R</a>	1054
7.84.1.25	<a href="#">nppiNormL2GetBufferHostSize_16s_AC4R</a>	1055
7.84.1.26	<a href="#">nppiNormL2GetBufferHostSize_16s_C1R</a>	1055

7.84.1.27	<a href="#">nppiNormL2GetBufferHostSize_16s_C3R</a>	1055
7.84.1.28	<a href="#">nppiNormL2GetBufferHostSize_16s_C4R</a>	1055
7.84.1.29	<a href="#">nppiNormL2GetBufferHostSize_16u_AC4R</a>	1056
7.84.1.30	<a href="#">nppiNormL2GetBufferHostSize_16u_C1MR</a>	1056
7.84.1.31	<a href="#">nppiNormL2GetBufferHostSize_16u_C1R</a>	1056
7.84.1.32	<a href="#">nppiNormL2GetBufferHostSize_16u_C3CMR</a>	1057
7.84.1.33	<a href="#">nppiNormL2GetBufferHostSize_16u_C3R</a>	1057
7.84.1.34	<a href="#">nppiNormL2GetBufferHostSize_16u_C4R</a>	1057
7.84.1.35	<a href="#">nppiNormL2GetBufferHostSize_32f_AC4R</a>	1057
7.84.1.36	<a href="#">nppiNormL2GetBufferHostSize_32f_C1MR</a>	1058
7.84.1.37	<a href="#">nppiNormL2GetBufferHostSize_32f_C1R</a>	1058
7.84.1.38	<a href="#">nppiNormL2GetBufferHostSize_32f_C3CMR</a>	1058
7.84.1.39	<a href="#">nppiNormL2GetBufferHostSize_32f_C3R</a>	1059
7.84.1.40	<a href="#">nppiNormL2GetBufferHostSize_32f_C4R</a>	1059
7.84.1.41	<a href="#">nppiNormL2GetBufferHostSize_8s_C1MR</a>	1059
7.84.1.42	<a href="#">nppiNormL2GetBufferHostSize_8s_C3CMR</a>	1059
7.84.1.43	<a href="#">nppiNormL2GetBufferHostSize_8u_AC4R</a>	1060
7.84.1.44	<a href="#">nppiNormL2GetBufferHostSize_8u_C1MR</a>	1060
7.84.1.45	<a href="#">nppiNormL2GetBufferHostSize_8u_C1R</a>	1060
7.84.1.46	<a href="#">nppiNormL2GetBufferHostSize_8u_C3CMR</a>	1061
7.84.1.47	<a href="#">nppiNormL2GetBufferHostSize_8u_C3R</a>	1061
7.84.1.48	<a href="#">nppiNormL2GetBufferHostSize_8u_C4R</a>	1061
7.85	<a href="#">Norm Diff</a>	1062
7.85.1	<a href="#">Function Documentation</a>	1062
7.85.1.1	<a href="#">nppiNormDiff_Inf_8u_C1R</a>	1062
7.85.1.2	<a href="#">nppiNormDiff_L1_8u_C1R</a>	1062
7.85.1.3	<a href="#">nppiNormDiff_L2_8u_C1R</a>	1063
7.86	<a href="#">Integral and Rectangular Standard Deviation</a>	1064
7.86.1	<a href="#">Function Documentation</a>	1064
7.86.1.1	<a href="#">nppiRectStdDev_32s32f_C1R</a>	1064
7.86.1.2	<a href="#">nppiSqrIntegral_8u32s32f_C1R</a>	1064
7.87	<a href="#">Histogram</a>	1066
7.87.1	<a href="#">Function Documentation</a>	1070
7.87.1.1	<a href="#">nppiEvenLevelsHost_32s</a>	1070
7.87.1.2	<a href="#">nppiHistogramEven_16s_AC4R</a>	1071
7.87.1.3	<a href="#">nppiHistogramEven_16s_C1R</a>	1071

7.87.1.4	nppiHistogramEven_16s_C3R	1072
7.87.1.5	nppiHistogramEven_16s_C4R	1072
7.87.1.6	nppiHistogramEven_16u_AC4R	1073
7.87.1.7	nppiHistogramEven_16u_C1R	1073
7.87.1.8	nppiHistogramEven_16u_C3R	1074
7.87.1.9	nppiHistogramEven_16u_C4R	1074
7.87.1.10	nppiHistogramEven_8u_AC4R	1075
7.87.1.11	nppiHistogramEven_8u_C1R	1075
7.87.1.12	nppiHistogramEven_8u_C3R	1076
7.87.1.13	nppiHistogramEven_8u_C4R	1076
7.87.1.14	nppiHistogramEvenGetBufferSize_16s_AC4R	1077
7.87.1.15	nppiHistogramEvenGetBufferSize_16s_C1R	1077
7.87.1.16	nppiHistogramEvenGetBufferSize_16s_C3R	1077
7.87.1.17	nppiHistogramEvenGetBufferSize_16s_C4R	1077
7.87.1.18	nppiHistogramEvenGetBufferSize_16u_AC4R	1078
7.87.1.19	nppiHistogramEvenGetBufferSize_16u_C1R	1078
7.87.1.20	nppiHistogramEvenGetBufferSize_16u_C3R	1078
7.87.1.21	nppiHistogramEvenGetBufferSize_16u_C4R	1079
7.87.1.22	nppiHistogramEvenGetBufferSize_8u_AC4R	1079
7.87.1.23	nppiHistogramEvenGetBufferSize_8u_C1R	1079
7.87.1.24	nppiHistogramEvenGetBufferSize_8u_C3R	1079
7.87.1.25	nppiHistogramEvenGetBufferSize_8u_C4R	1080
7.87.1.26	nppiHistogramRange_16s_AC4R	1080
7.87.1.27	nppiHistogramRange_16s_C1R	1081
7.87.1.28	nppiHistogramRange_16s_C3R	1081
7.87.1.29	nppiHistogramRange_16s_C4R	1081
7.87.1.30	nppiHistogramRange_16u_AC4R	1082
7.87.1.31	nppiHistogramRange_16u_C1R	1082
7.87.1.32	nppiHistogramRange_16u_C3R	1083
7.87.1.33	nppiHistogramRange_16u_C4R	1083
7.87.1.34	nppiHistogramRange_32f_AC4R	1084
7.87.1.35	nppiHistogramRange_32f_C1R	1084
7.87.1.36	nppiHistogramRange_32f_C3R	1085
7.87.1.37	nppiHistogramRange_32f_C4R	1085
7.87.1.38	nppiHistogramRange_8u_AC4R	1085
7.87.1.39	nppiHistogramRange_8u_C1R	1086

7.87.1.40	<a href="#">nppiHistogramRange_8u_C3R</a>	1086
7.87.1.41	<a href="#">nppiHistogramRange_8u_C4R</a>	1087
7.87.1.42	<a href="#">nppiHistogramRangeGetBufferSize_16s_AC4R</a>	1087
7.87.1.43	<a href="#">nppiHistogramRangeGetBufferSize_16s_C1R</a>	1088
7.87.1.44	<a href="#">nppiHistogramRangeGetBufferSize_16s_C3R</a>	1088
7.87.1.45	<a href="#">nppiHistogramRangeGetBufferSize_16s_C4R</a>	1088
7.87.1.46	<a href="#">nppiHistogramRangeGetBufferSize_16u_AC4R</a>	1088
7.87.1.47	<a href="#">nppiHistogramRangeGetBufferSize_16u_C1R</a>	1089
7.87.1.48	<a href="#">nppiHistogramRangeGetBufferSize_16u_C3R</a>	1089
7.87.1.49	<a href="#">nppiHistogramRangeGetBufferSize_16u_C4R</a>	1089
7.87.1.50	<a href="#">nppiHistogramRangeGetBufferSize_32f_AC4R</a>	1090
7.87.1.51	<a href="#">nppiHistogramRangeGetBufferSize_32f_C1R</a>	1090
7.87.1.52	<a href="#">nppiHistogramRangeGetBufferSize_32f_C3R</a>	1090
7.87.1.53	<a href="#">nppiHistogramRangeGetBufferSize_32f_C4R</a>	1090
7.87.1.54	<a href="#">nppiHistogramRangeGetBufferSize_8u_AC4R</a>	1091
7.87.1.55	<a href="#">nppiHistogramRangeGetBufferSize_8u_C1R</a>	1091
7.87.1.56	<a href="#">nppiHistogramRangeGetBufferSize_8u_C3R</a>	1091
7.87.1.57	<a href="#">nppiHistogramRangeGetBufferSize_8u_C4R</a>	1092
7.88	<a href="#">Memory Management</a>	1093
7.88.1	<a href="#">Detailed Description</a>	1095
7.88.2	<a href="#">Function Documentation</a>	1095
7.88.2.1	<a href="#">nppiFree</a>	1095
7.88.2.2	<a href="#">nppiMalloc_16s_C1</a>	1095
7.88.2.3	<a href="#">nppiMalloc_16s_C2</a>	1096
7.88.2.4	<a href="#">nppiMalloc_16s_C4</a>	1096
7.88.2.5	<a href="#">nppiMalloc_16sc_C1</a>	1096
7.88.2.6	<a href="#">nppiMalloc_16sc_C2</a>	1096
7.88.2.7	<a href="#">nppiMalloc_16sc_C3</a>	1097
7.88.2.8	<a href="#">nppiMalloc_16sc_C4</a>	1097
7.88.2.9	<a href="#">nppiMalloc_16u_C1</a>	1097
7.88.2.10	<a href="#">nppiMalloc_16u_C2</a>	1098
7.88.2.11	<a href="#">nppiMalloc_16u_C3</a>	1098
7.88.2.12	<a href="#">nppiMalloc_16u_C4</a>	1098
7.88.2.13	<a href="#">nppiMalloc_32f_C1</a>	1098
7.88.2.14	<a href="#">nppiMalloc_32f_C2</a>	1099
7.88.2.15	<a href="#">nppiMalloc_32f_C3</a>	1099

7.88.2.16	nppiMalloc_32f_C4	1099
7.88.2.17	nppiMalloc_32fc_C1	1100
7.88.2.18	nppiMalloc_32fc_C2	1100
7.88.2.19	nppiMalloc_32fc_C3	1100
7.88.2.20	nppiMalloc_32fc_C4	1100
7.88.2.21	nppiMalloc_32s_C1	1101
7.88.2.22	nppiMalloc_32s_C3	1101
7.88.2.23	nppiMalloc_32s_C4	1101
7.88.2.24	nppiMalloc_32sc_C1	1102
7.88.2.25	nppiMalloc_32sc_C2	1102
7.88.2.26	nppiMalloc_32sc_C3	1102
7.88.2.27	nppiMalloc_32sc_C4	1102
7.88.2.28	nppiMalloc_8u_C1	1103
7.88.2.29	nppiMalloc_8u_C2	1103
7.88.2.30	nppiMalloc_8u_C3	1103
7.88.2.31	nppiMalloc_8u_C4	1104
7.89	Threshold and Compare Operations	1105
7.89.1	Detailed Description	1105
7.90	Threshold Operations	1106
7.90.1	Detailed Description	1120
7.90.2	Function Documentation	1120
7.90.2.1	nppiThreshold_16s_AC4IR	1120
7.90.2.2	nppiThreshold_16s_AC4R	1120
7.90.2.3	nppiThreshold_16s_C1IR	1121
7.90.2.4	nppiThreshold_16s_C1R	1121
7.90.2.5	nppiThreshold_16s_C3IR	1122
7.90.2.6	nppiThreshold_16s_C3R	1122
7.90.2.7	nppiThreshold_16u_AC4IR	1123
7.90.2.8	nppiThreshold_16u_AC4R	1123
7.90.2.9	nppiThreshold_16u_C1IR	1124
7.90.2.10	nppiThreshold_16u_C1R	1124
7.90.2.11	nppiThreshold_16u_C3IR	1124
7.90.2.12	nppiThreshold_16u_C3R	1125
7.90.2.13	nppiThreshold_32f_AC4IR	1125
7.90.2.14	nppiThreshold_32f_AC4R	1126
7.90.2.15	nppiThreshold_32f_C1IR	1126

---

7.90.2.16	nppiThreshold_32f_C1R	1127
7.90.2.17	nppiThreshold_32f_C3IR	1127
7.90.2.18	nppiThreshold_32f_C3R	1128
7.90.2.19	nppiThreshold_8u_AC4IR	1128
7.90.2.20	nppiThreshold_8u_AC4R	1129
7.90.2.21	nppiThreshold_8u_C1IR	1129
7.90.2.22	nppiThreshold_8u_C1R	1130
7.90.2.23	nppiThreshold_8u_C3IR	1130
7.90.2.24	nppiThreshold_8u_C3R	1131
7.90.2.25	nppiThreshold_GT_16s_AC4IR	1131
7.90.2.26	nppiThreshold_GT_16s_AC4R	1131
7.90.2.27	nppiThreshold_GT_16s_C1IR	1132
7.90.2.28	nppiThreshold_GT_16s_C1R	1132
7.90.2.29	nppiThreshold_GT_16s_C3IR	1133
7.90.2.30	nppiThreshold_GT_16s_C3R	1133
7.90.2.31	nppiThreshold_GT_16u_AC4IR	1133
7.90.2.32	nppiThreshold_GT_16u_AC4R	1134
7.90.2.33	nppiThreshold_GT_16u_C1IR	1134
7.90.2.34	nppiThreshold_GT_16u_C1R	1135
7.90.2.35	nppiThreshold_GT_16u_C3IR	1135
7.90.2.36	nppiThreshold_GT_16u_C3R	1135
7.90.2.37	nppiThreshold_GT_32f_AC4IR	1136
7.90.2.38	nppiThreshold_GT_32f_AC4R	1136
7.90.2.39	nppiThreshold_GT_32f_C1IR	1137
7.90.2.40	nppiThreshold_GT_32f_C1R	1137
7.90.2.41	nppiThreshold_GT_32f_C3IR	1137
7.90.2.42	nppiThreshold_GT_32f_C3R	1138
7.90.2.43	nppiThreshold_GT_8u_AC4IR	1138
7.90.2.44	nppiThreshold_GT_8u_AC4R	1139
7.90.2.45	nppiThreshold_GT_8u_C1IR	1139
7.90.2.46	nppiThreshold_GT_8u_C1R	1139
7.90.2.47	nppiThreshold_GT_8u_C3IR	1140
7.90.2.48	nppiThreshold_GT_8u_C3R	1140
7.90.2.49	nppiThreshold_GTVVal_16s_AC4IR	1141
7.90.2.50	nppiThreshold_GTVVal_16s_AC4R	1141
7.90.2.51	nppiThreshold_GTVVal_16s_C1IR	1141

---

7.90.2.52	nppiThreshold_GTVal_16s_C1R	1142
7.90.2.53	nppiThreshold_GTVal_16s_C3IR	1142
7.90.2.54	nppiThreshold_GTVal_16s_C3R	1143
7.90.2.55	nppiThreshold_GTVal_16u_AC4IR	1143
7.90.2.56	nppiThreshold_GTVal_16u_AC4R	1143
7.90.2.57	nppiThreshold_GTVal_16u_C1IR	1144
7.90.2.58	nppiThreshold_GTVal_16u_C1R	1144
7.90.2.59	nppiThreshold_GTVal_16u_C3IR	1145
7.90.2.60	nppiThreshold_GTVal_16u_C3R	1145
7.90.2.61	nppiThreshold_GTVal_32f_AC4IR	1146
7.90.2.62	nppiThreshold_GTVal_32f_AC4R	1146
7.90.2.63	nppiThreshold_GTVal_32f_C1IR	1146
7.90.2.64	nppiThreshold_GTVal_32f_C1R	1147
7.90.2.65	nppiThreshold_GTVal_32f_C3IR	1147
7.90.2.66	nppiThreshold_GTVal_32f_C3R	1148
7.90.2.67	nppiThreshold_GTVal_8u_AC4IR	1148
7.90.2.68	nppiThreshold_GTVal_8u_AC4R	1148
7.90.2.69	nppiThreshold_GTVal_8u_C1IR	1149
7.90.2.70	nppiThreshold_GTVal_8u_C1R	1149
7.90.2.71	nppiThreshold_GTVal_8u_C3IR	1150
7.90.2.72	nppiThreshold_GTVal_8u_C3R	1150
7.90.2.73	nppiThreshold_LT_16s_AC4IR	1151
7.90.2.74	nppiThreshold_LT_16s_AC4R	1151
7.90.2.75	nppiThreshold_LT_16s_C1IR	1151
7.90.2.76	nppiThreshold_LT_16s_C1R	1152
7.90.2.77	nppiThreshold_LT_16s_C3IR	1152
7.90.2.78	nppiThreshold_LT_16s_C3R	1153
7.90.2.79	nppiThreshold_LT_16u_AC4IR	1153
7.90.2.80	nppiThreshold_LT_16u_AC4R	1153
7.90.2.81	nppiThreshold_LT_16u_C1IR	1154
7.90.2.82	nppiThreshold_LT_16u_C1R	1154
7.90.2.83	nppiThreshold_LT_16u_C3IR	1155
7.90.2.84	nppiThreshold_LT_16u_C3R	1155
7.90.2.85	nppiThreshold_LT_32f_AC4IR	1155
7.90.2.86	nppiThreshold_LT_32f_AC4R	1156
7.90.2.87	nppiThreshold_LT_32f_C1IR	1156

7.90.2.88	nppiThreshold_LT_32f_C1R	1157
7.90.2.89	nppiThreshold_LT_32f_C3IR	1157
7.90.2.90	nppiThreshold_LT_32f_C3R	1157
7.90.2.91	nppiThreshold_LT_8u_AC4IR	1158
7.90.2.92	nppiThreshold_LT_8u_AC4R	1158
7.90.2.93	nppiThreshold_LT_8u_C1IR	1159
7.90.2.94	nppiThreshold_LT_8u_C1R	1159
7.90.2.95	nppiThreshold_LT_8u_C3IR	1159
7.90.2.96	nppiThreshold_LT_8u_C3R	1160
7.90.2.97	nppiThreshold_LTVa1_16s_AC4IR	1160
7.90.2.98	nppiThreshold_LTVa1_16s_AC4R	1161
7.90.2.99	nppiThreshold_LTVa1_16s_C1IR	1161
7.90.2.100	nppiThreshold_LTVa1_16s_C1R	1161
7.90.2.101	nppiThreshold_LTVa1_16s_C3IR	1162
7.90.2.102	nppiThreshold_LTVa1_16s_C3R	1162
7.90.2.103	nppiThreshold_LTVa1_16u_AC4IR	1163
7.90.2.104	nppiThreshold_LTVa1_16u_AC4R	1163
7.90.2.105	nppiThreshold_LTVa1_16u_C1IR	1164
7.90.2.106	nppiThreshold_LTVa1_16u_C1R	1164
7.90.2.107	nppiThreshold_LTVa1_16u_C3IR	1164
7.90.2.108	nppiThreshold_LTVa1_16u_C3R	1165
7.90.2.109	nppiThreshold_LTVa1_32f_AC4IR	1165
7.90.2.110	nppiThreshold_LTVa1_32f_AC4R	1166
7.90.2.111	nppiThreshold_LTVa1_32f_C1IR	1166
7.90.2.112	nppiThreshold_LTVa1_32f_C1R	1166
7.90.2.113	nppiThreshold_LTVa1_32f_C3IR	1167
7.90.2.114	nppiThreshold_LTVa1_32f_C3R	1167
7.90.2.115	nppiThreshold_LTVa1_8u_AC4IR	1168
7.90.2.116	nppiThreshold_LTVa1_8u_AC4R	1168
7.90.2.117	nppiThreshold_LTVa1_8u_C1IR	1169
7.90.2.118	nppiThreshold_LTVa1_8u_C1R	1169
7.90.2.119	nppiThreshold_LTVa1_8u_C3IR	1169
7.90.2.120	nppiThreshold_LTVa1_8u_C3R	1170
7.90.2.121	nppiThreshold_LTVa1GTVal_16s_AC4IR	1170
7.90.2.122	nppiThreshold_LTVa1GTVal_16s_AC4R	1171
7.90.2.123	nppiThreshold_LTVa1GTVal_16s_C1IR	1171

7.90.2.124nppiThreshold_LTVaGTVal_16s_C1R . . . . .	1172
7.90.2.125nppiThreshold_LTVaGTVal_16s_C3IR . . . . .	1172
7.90.2.126nppiThreshold_LTVaGTVal_16s_C3R . . . . .	1173
7.90.2.127nppiThreshold_LTVaGTVal_16u_AC4IR . . . . .	1173
7.90.2.128nppiThreshold_LTVaGTVal_16u_AC4R . . . . .	1174
7.90.2.129nppiThreshold_LTVaGTVal_16u_C1IR . . . . .	1174
7.90.2.130nppiThreshold_LTVaGTVal_16u_C1R . . . . .	1175
7.90.2.131nppiThreshold_LTVaGTVal_16u_C3IR . . . . .	1175
7.90.2.132nppiThreshold_LTVaGTVal_16u_C3R . . . . .	1176
7.90.2.133nppiThreshold_LTVaGTVal_32f_AC4IR . . . . .	1176
7.90.2.134nppiThreshold_LTVaGTVal_32f_AC4R . . . . .	1177
7.90.2.135nppiThreshold_LTVaGTVal_32f_C1IR . . . . .	1177
7.90.2.136nppiThreshold_LTVaGTVal_32f_C1R . . . . .	1178
7.90.2.137nppiThreshold_LTVaGTVal_32f_C3IR . . . . .	1178
7.90.2.138nppiThreshold_LTVaGTVal_32f_C3R . . . . .	1179
7.90.2.139nppiThreshold_LTVaGTVal_8u_AC4IR . . . . .	1179
7.90.2.140nppiThreshold_LTVaGTVal_8u_AC4R . . . . .	1180
7.90.2.141nppiThreshold_LTVaGTVal_8u_C1IR . . . . .	1180
7.90.2.142nppiThreshold_LTVaGTVal_8u_C1R . . . . .	1181
7.90.2.143nppiThreshold_LTVaGTVal_8u_C3IR . . . . .	1181
7.90.2.144nppiThreshold_LTVaGTVal_8u_C3R . . . . .	1182
7.90.2.145nppiThreshold_Val_16s_AC4IR . . . . .	1182
7.90.2.146nppiThreshold_Val_16s_AC4R . . . . .	1183
7.90.2.147nppiThreshold_Val_16s_C1IR . . . . .	1183
7.90.2.148nppiThreshold_Val_16s_C1R . . . . .	1184
7.90.2.149nppiThreshold_Val_16s_C3IR . . . . .	1184
7.90.2.150nppiThreshold_Val_16s_C3R . . . . .	1185
7.90.2.151nppiThreshold_Val_16u_AC4IR . . . . .	1185
7.90.2.152nppiThreshold_Val_16u_AC4R . . . . .	1186
7.90.2.153nppiThreshold_Val_16u_C1IR . . . . .	1186
7.90.2.154nppiThreshold_Val_16u_C1R . . . . .	1187
7.90.2.155nppiThreshold_Val_16u_C3IR . . . . .	1187
7.90.2.156nppiThreshold_Val_16u_C3R . . . . .	1188
7.90.2.157nppiThreshold_Val_32f_AC4IR . . . . .	1188
7.90.2.158nppiThreshold_Val_32f_AC4R . . . . .	1189
7.90.2.159nppiThreshold_Val_32f_C1IR . . . . .	1189

7.90.2.160	nppiThreshold_Val_32f_C1R	1190
7.90.2.161	nppiThreshold_Val_32f_C3IR	1190
7.90.2.162	nppiThreshold_Val_32f_C3R	1191
7.90.2.163	nppiThreshold_Val_8u_AC4IR	1191
7.90.2.164	nppiThreshold_Val_8u_AC4R	1192
7.90.2.165	nppiThreshold_Val_8u_C1IR	1192
7.90.2.166	nppiThreshold_Val_8u_C1R	1193
7.90.2.167	nppiThreshold_Val_8u_C3IR	1193
7.90.2.168	nppiThreshold_Val_8u_C3R	1194
7.91	Compare Operations	1195
7.91.1	Detailed Description	1198
7.91.2	Function Documentation	1198
7.91.2.1	nppiCompare_16s_AC4R	1198
7.91.2.2	nppiCompare_16s_C1R	1199
7.91.2.3	nppiCompare_16s_C3R	1199
7.91.2.4	nppiCompare_16s_C4R	1200
7.91.2.5	nppiCompare_16u_AC4R	1200
7.91.2.6	nppiCompare_16u_C1R	1201
7.91.2.7	nppiCompare_16u_C3R	1201
7.91.2.8	nppiCompare_16u_C4R	1202
7.91.2.9	nppiCompare_32f_AC4R	1202
7.91.2.10	nppiCompare_32f_C1R	1203
7.91.2.11	nppiCompare_32f_C3R	1203
7.91.2.12	nppiCompare_32f_C4R	1204
7.91.2.13	nppiCompare_8u_AC4R	1204
7.91.2.14	nppiCompare_8u_C1R	1205
7.91.2.15	nppiCompare_8u_C3R	1205
7.91.2.16	nppiCompare_8u_C4R	1206
7.91.2.17	nppiCompareC_16s_AC4R	1206
7.91.2.18	nppiCompareC_16s_C1R	1206
7.91.2.19	nppiCompareC_16s_C3R	1207
7.91.2.20	nppiCompareC_16s_C4R	1207
7.91.2.21	nppiCompareC_16u_AC4R	1208
7.91.2.22	nppiCompareC_16u_C1R	1208
7.91.2.23	nppiCompareC_16u_C3R	1209
7.91.2.24	nppiCompareC_16u_C4R	1209

7.91.2.25	<code>nppiCompareC_32f_AC4R</code>	1209
7.91.2.26	<code>nppiCompareC_32f_C1R</code>	1210
7.91.2.27	<code>nppiCompareC_32f_C3R</code>	1210
7.91.2.28	<code>nppiCompareC_32f_C4R</code>	1211
7.91.2.29	<code>nppiCompareC_8u_AC4R</code>	1211
7.91.2.30	<code>nppiCompareC_8u_C1R</code>	1212
7.91.2.31	<code>nppiCompareC_8u_C3R</code>	1212
7.91.2.32	<code>nppiCompareC_8u_C4R</code>	1212
7.91.2.33	<code>nppiCompareEqualEps_32f_AC4R</code>	1213
7.91.2.34	<code>nppiCompareEqualEps_32f_C1R</code>	1213
7.91.2.35	<code>nppiCompareEqualEps_32f_C3R</code>	1214
7.91.2.36	<code>nppiCompareEqualEps_32f_C4R</code>	1214
7.91.2.37	<code>nppiCompareEqualEpsC_32f_AC4R</code>	1215
7.91.2.38	<code>nppiCompareEqualEpsC_32f_C1R</code>	1215
7.91.2.39	<code>nppiCompareEqualEpsC_32f_C3R</code>	1216
7.91.2.40	<code>nppiCompareEqualEpsC_32f_C4R</code>	1216
7.92	NPP Signal Processing	1217
7.93	Arithmetic and Logical Operations	1218
7.94	Arithmetic Operations	1219
7.95	AddC	1221
7.95.1	Detailed Description	1222
7.95.2	Function Documentation	1222
7.95.2.1	<code>nppsAddC_16s_ISfs</code>	1222
7.95.2.2	<code>nppsAddC_16s_Sfs</code>	1223
7.95.2.3	<code>nppsAddC_16sc_ISfs</code>	1223
7.95.2.4	<code>nppsAddC_16sc_Sfs</code>	1223
7.95.2.5	<code>nppsAddC_16u_ISfs</code>	1224
7.95.2.6	<code>nppsAddC_16u_Sfs</code>	1224
7.95.2.7	<code>nppsAddC_32f</code>	1224
7.95.2.8	<code>nppsAddC_32f_I</code>	1225
7.95.2.9	<code>nppsAddC_32fc</code>	1225
7.95.2.10	<code>nppsAddC_32fc_I</code>	1225
7.95.2.11	<code>nppsAddC_32s_ISfs</code>	1225
7.95.2.12	<code>nppsAddC_32s_Sfs</code>	1226
7.95.2.13	<code>nppsAddC_32sc_ISfs</code>	1226
7.95.2.14	<code>nppsAddC_32sc_Sfs</code>	1227

7.95.2.15	nppsAddC_64f	1227
7.95.2.16	nppsAddC_64f_I	1227
7.95.2.17	nppsAddC_64fc	1228
7.95.2.18	nppsAddC_64fc_I	1228
7.95.2.19	nppsAddC_8u_ISfs	1228
7.95.2.20	nppsAddC_8u_Sfs	1229
7.96	AddProductC	1230
7.96.1	Detailed Description	1230
7.96.2	Function Documentation	1230
7.96.2.1	nppsAddProductC_32f	1230
7.97	MulC	1231
7.97.1	Detailed Description	1232
7.97.2	Function Documentation	1232
7.97.2.1	nppsMulC_16s_ISfs	1232
7.97.2.2	nppsMulC_16s_Sfs	1233
7.97.2.3	nppsMulC_16sc_ISfs	1233
7.97.2.4	nppsMulC_16sc_Sfs	1234
7.97.2.5	nppsMulC_16u_ISfs	1234
7.97.2.6	nppsMulC_16u_Sfs	1234
7.97.2.7	nppsMulC_32f	1235
7.97.2.8	nppsMulC_32f16s_Sfs	1235
7.97.2.9	nppsMulC_32f_I	1235
7.97.2.10	nppsMulC_32fc	1236
7.97.2.11	nppsMulC_32fc_I	1236
7.97.2.12	nppsMulC_32s_ISfs	1236
7.97.2.13	nppsMulC_32s_Sfs	1237
7.97.2.14	nppsMulC_32sc_ISfs	1237
7.97.2.15	nppsMulC_32sc_Sfs	1237
7.97.2.16	nppsMulC_64f	1238
7.97.2.17	nppsMulC_64f64s_ISfs	1238
7.97.2.18	nppsMulC_64f_I	1238
7.97.2.19	nppsMulC_64fc	1239
7.97.2.20	nppsMulC_64fc_I	1239
7.97.2.21	nppsMulC_8u_ISfs	1239
7.97.2.22	nppsMulC_8u_Sfs	1240
7.97.2.23	nppsMulC_Low_32f16s	1240

---

7.98 SubC	1241
7.98.1 Detailed Description	1242
7.98.2 Function Documentation	1242
7.98.2.1 nppsSubC_16s_ISfs	1242
7.98.2.2 nppsSubC_16s_Sfs	1243
7.98.2.3 nppsSubC_16sc_ISfs	1243
7.98.2.4 nppsSubC_16sc_Sfs	1243
7.98.2.5 nppsSubC_16u_ISfs	1244
7.98.2.6 nppsSubC_16u_Sfs	1244
7.98.2.7 nppsSubC_32f	1244
7.98.2.8 nppsSubC_32f_I	1245
7.98.2.9 nppsSubC_32fc	1245
7.98.2.10 nppsSubC_32fc_I	1245
7.98.2.11 nppsSubC_32s_ISfs	1245
7.98.2.12 nppsSubC_32s_Sfs	1246
7.98.2.13 nppsSubC_32sc_ISfs	1246
7.98.2.14 nppsSubC_32sc_Sfs	1247
7.98.2.15 nppsSubC_64f	1247
7.98.2.16 nppsSubC_64f_I	1247
7.98.2.17 nppsSubC_64fc	1248
7.98.2.18 nppsSubC_64fc_I	1248
7.98.2.19 nppsSubC_8u_ISfs	1248
7.98.2.20 nppsSubC_8u_Sfs	1249
7.99 SubCRev	1250
7.99.1 Detailed Description	1251
7.99.2 Function Documentation	1251
7.99.2.1 nppsSubCRev_16s_ISfs	1251
7.99.2.2 nppsSubCRev_16s_Sfs	1252
7.99.2.3 nppsSubCRev_16sc_ISfs	1252
7.99.2.4 nppsSubCRev_16sc_Sfs	1252
7.99.2.5 nppsSubCRev_16u_ISfs	1253
7.99.2.6 nppsSubCRev_16u_Sfs	1253
7.99.2.7 nppsSubCRev_32f	1253
7.99.2.8 nppsSubCRev_32f_I	1254
7.99.2.9 nppsSubCRev_32fc	1254
7.99.2.10 nppsSubCRev_32fc_I	1254

7.99.2.11 nppsSubCRev_32s_ISfs . . . . .	1255
7.99.2.12 nppsSubCRev_32s_Sfs . . . . .	1255
7.99.2.13 nppsSubCRev_32sc_ISfs . . . . .	1255
7.99.2.14 nppsSubCRev_32sc_Sfs . . . . .	1256
7.99.2.15 nppsSubCRev_64f . . . . .	1256
7.99.2.16 nppsSubCRev_64f_I . . . . .	1256
7.99.2.17 nppsSubCRev_64fc . . . . .	1257
7.99.2.18 nppsSubCRev_64fc_I . . . . .	1257
7.99.2.19 nppsSubCRev_8u_ISfs . . . . .	1257
7.99.2.20 nppsSubCRev_8u_Sfs . . . . .	1258
7.100DivC . . . . .	1259
7.100.1 Detailed Description . . . . .	1260
7.100.2 Function Documentation . . . . .	1260
7.100.2.1 nppsDivC_16s_ISfs . . . . .	1260
7.100.2.2 nppsDivC_16s_Sfs . . . . .	1260
7.100.2.3 nppsDivC_16sc_ISfs . . . . .	1261
7.100.2.4 nppsDivC_16sc_Sfs . . . . .	1261
7.100.2.5 nppsDivC_16u_ISfs . . . . .	1261
7.100.2.6 nppsDivC_16u_Sfs . . . . .	1262
7.100.2.7 nppsDivC_32f . . . . .	1262
7.100.2.8 nppsDivC_32f_I . . . . .	1262
7.100.2.9 nppsDivC_32fc . . . . .	1263
7.100.2.10 nppsDivC_32fc_I . . . . .	1263
7.100.2.11 nppsDivC_64f . . . . .	1263
7.100.2.12 nppsDivC_64f_I . . . . .	1264
7.100.2.13 nppsDivC_64fc . . . . .	1264
7.100.2.14 nppsDivC_64fc_I . . . . .	1264
7.100.2.15 nppsDivC_8u_ISfs . . . . .	1264
7.100.2.16 nppsDivC_8u_Sfs . . . . .	1265
7.101DivCRev . . . . .	1266
7.101.1 Detailed Description . . . . .	1266
7.101.2 Function Documentation . . . . .	1266
7.101.2.1 nppsDivCRev_16u . . . . .	1266
7.101.2.2 nppsDivCRev_16u_I . . . . .	1266
7.101.2.3 nppsDivCRev_32f . . . . .	1267
7.101.2.4 nppsDivCRev_32f_I . . . . .	1267

---

7.102Add	1268
7.102.1 Detailed Description	1270
7.102.2 Function Documentation	1270
7.102.2.1 nppsAdd_16s	1270
7.102.2.2 nppsAdd_16s32f	1270
7.102.2.3 nppsAdd_16s32s_I	1271
7.102.2.4 nppsAdd_16s_I	1271
7.102.2.5 nppsAdd_16s_ISfs	1271
7.102.2.6 nppsAdd_16s_Sfs	1272
7.102.2.7 nppsAdd_16sc_ISfs	1272
7.102.2.8 nppsAdd_16sc_Sfs	1272
7.102.2.9 nppsAdd_16u	1273
7.102.2.10 nppsAdd_16u_ISfs	1273
7.102.2.11 nppsAdd_16u_Sfs	1273
7.102.2.12 nppsAdd_32f	1274
7.102.2.13 nppsAdd_32f_I	1274
7.102.2.14 nppsAdd_32fc	1274
7.102.2.15 nppsAdd_32fc_I	1275
7.102.2.16 nppsAdd_32s_ISfs	1275
7.102.2.17 nppsAdd_32s_Sfs	1275
7.102.2.18 nppsAdd_32sc_ISfs	1276
7.102.2.19 nppsAdd_32sc_Sfs	1276
7.102.2.20 nppsAdd_32u	1276
7.102.2.21 nppsAdd_64f	1277
7.102.2.22 nppsAdd_64f_I	1277
7.102.2.23 nppsAdd_64fc	1277
7.102.2.24 nppsAdd_64fc_I	1278
7.102.2.25 nppsAdd_64s_Sfs	1278
7.102.2.26 nppsAdd_8u16u	1278
7.102.2.27 nppsAdd_8u_ISfs	1279
7.102.2.28 nppsAdd_8u_Sfs	1279
7.103AddProduct	1280
7.103.1 Detailed Description	1280
7.103.2 Function Documentation	1281
7.103.2.1 nppsAddProduct_16s32s_Sfs	1281
7.103.2.2 nppsAddProduct_16s_Sfs	1281

7.103.2.3 nppsAddProduct_32f	1281
7.103.2.4 nppsAddProduct_32fc	1282
7.103.2.5 nppsAddProduct_32s_Sfs	1282
7.103.2.6 nppsAddProduct_64f	1283
7.103.2.7 nppsAddProduct_64fc	1283
7.104Mul	1284
7.104.1 Detailed Description	1286
7.104.2 Function Documentation	1286
7.104.2.1 nppsMul_16s	1286
7.104.2.2 nppsMul_16s32f	1287
7.104.2.3 nppsMul_16s32s_Sfs	1287
7.104.2.4 nppsMul_16s_I	1287
7.104.2.5 nppsMul_16s_ISfs	1288
7.104.2.6 nppsMul_16s_Sfs	1288
7.104.2.7 nppsMul_16sc_ISfs	1288
7.104.2.8 nppsMul_16sc_Sfs	1289
7.104.2.9 nppsMul_16u16s_Sfs	1289
7.104.2.10nppsMul_16u_ISfs	1289
7.104.2.11nppsMul_16u_Sfs	1290
7.104.2.12nppsMul_32f	1290
7.104.2.13nppsMul_32f32fc	1290
7.104.2.14nppsMul_32f32fc_I	1291
7.104.2.15nppsMul_32f_I	1291
7.104.2.16nppsMul_32fc	1291
7.104.2.17nppsMul_32fc_I	1292
7.104.2.18nppsMul_32s32sc_ISfs	1292
7.104.2.19nppsMul_32s32sc_Sfs	1292
7.104.2.20nppsMul_32s_ISfs	1293
7.104.2.21nppsMul_32s_Sfs	1293
7.104.2.22nppsMul_32sc_ISfs	1293
7.104.2.23nppsMul_32sc_Sfs	1294
7.104.2.24nppsMul_64f	1294
7.104.2.25nppsMul_64f_I	1294
7.104.2.26nppsMul_64fc	1295
7.104.2.27nppsMul_64fc_I	1295
7.104.2.28nppsMul_8u16u	1295

7.104.2.29	nppsMul_8u_ISfs	1296
7.104.2.30	nppsMul_8u_Sfs	1296
7.104.2.31	nppsMul_Low_32s_Sfs	1296
7.105	Sub	1297
7.105.1	Detailed Description	1298
7.105.2	Function Documentation	1298
7.105.2.1	nppsSub_16s	1298
7.105.2.2	nppsSub_16s32f	1299
7.105.2.3	nppsSub_16s_I	1299
7.105.2.4	nppsSub_16s_ISfs	1299
7.105.2.5	nppsSub_16s_Sfs	1300
7.105.2.6	nppsSub_16sc_ISfs	1300
7.105.2.7	nppsSub_16sc_Sfs	1300
7.105.2.8	nppsSub_16u_ISfs	1301
7.105.2.9	nppsSub_16u_Sfs	1301
7.105.2.10	nppsSub_32f	1301
7.105.2.11	nppsSub_32f_I	1302
7.105.2.12	nppsSub_32fc	1302
7.105.2.13	nppsSub_32fc_I	1302
7.105.2.14	nppsSub_32s_ISfs	1303
7.105.2.15	nppsSub_32s_Sfs	1303
7.105.2.16	nppsSub_32sc_ISfs	1303
7.105.2.17	nppsSub_32sc_Sfs	1304
7.105.2.18	nppsSub_64f	1304
7.105.2.19	nppsSub_64f_I	1304
7.105.2.20	nppsSub_64fc	1305
7.105.2.21	nppsSub_64fc_I	1305
7.105.2.22	nppsSub_8u_ISfs	1305
7.105.2.23	nppsSub_8u_Sfs	1306
7.106	Div	1307
7.106.1	Detailed Description	1308
7.106.2	Function Documentation	1308
7.106.2.1	nppsDiv_16s_ISfs	1308
7.106.2.2	nppsDiv_16s_Sfs	1309
7.106.2.3	nppsDiv_16sc_ISfs	1309
7.106.2.4	nppsDiv_16sc_Sfs	1309

7.106.2.5	nppsDiv_16u_ISfs	1310
7.106.2.6	nppsDiv_16u_Sfs	1310
7.106.2.7	nppsDiv_32f	1310
7.106.2.8	nppsDiv_32f_I	1311
7.106.2.9	nppsDiv_32fc	1311
7.106.2.10	nppsDiv_32fc_I	1311
7.106.2.11	nppsDiv_32s16s_Sfs	1311
7.106.2.12	nppsDiv_32s_ISfs	1312
7.106.2.13	nppsDiv_32s_Sfs	1312
7.106.2.14	nppsDiv_64f	1313
7.106.2.15	nppsDiv_64f_I	1313
7.106.2.16	nppsDiv_64fc	1313
7.106.2.17	nppsDiv_64fc_I	1314
7.106.2.18	nppsDiv_8u_ISfs	1314
7.106.2.19	nppsDiv_8u_Sfs	1314
7.107	Div_Round	1315
7.107.1	Detailed Description	1315
7.107.2	Function Documentation	1315
7.107.2.1	nppsDiv_Round_16s_ISfs	1315
7.107.2.2	nppsDiv_Round_16s_Sfs	1316
7.107.2.3	nppsDiv_Round_16u_ISfs	1316
7.107.2.4	nppsDiv_Round_16u_Sfs	1316
7.107.2.5	nppsDiv_Round_8u_ISfs	1317
7.107.2.6	nppsDiv_Round_8u_Sfs	1317
7.108	Abs	1318
7.108.1	Detailed Description	1318
7.108.2	Function Documentation	1318
7.108.2.1	nppsAbs_16s	1318
7.108.2.2	nppsAbs_16s_I	1319
7.108.2.3	nppsAbs_32f	1319
7.108.2.4	nppsAbs_32f_I	1319
7.108.2.5	nppsAbs_32s	1319
7.108.2.6	nppsAbs_32s_I	1320
7.108.2.7	nppsAbs_64f	1320
7.108.2.8	nppsAbs_64f_I	1320
7.109	Sqr	1321

---

7.109.1 Detailed Description	1322
7.109.2 Function Documentation	1322
7.109.2.1 nppsSqr_16s_ISfs	1322
7.109.2.2 nppsSqr_16s_Sfs	1322
7.109.2.3 nppsSqr_16sc_ISfs	1322
7.109.2.4 nppsSqr_16sc_Sfs	1323
7.109.2.5 nppsSqr_16u_ISfs	1323
7.109.2.6 nppsSqr_16u_Sfs	1323
7.109.2.7 nppsSqr_32f	1324
7.109.2.8 nppsSqr_32f_I	1324
7.109.2.9 nppsSqr_32fc	1324
7.109.2.10 nppsSqr_32fc_I	1324
7.109.2.11 nppsSqr_64f	1325
7.109.2.12 nppsSqr_64f_I	1325
7.109.2.13 nppsSqr_64fc	1325
7.109.2.14 nppsSqr_64fc_I	1325
7.109.2.15 nppsSqr_8u_ISfs	1326
7.109.2.16 nppsSqr_8u_Sfs	1326
7.110 Sqrt	1327
7.110.1 Detailed Description	1328
7.110.2 Function Documentation	1328
7.110.2.1 nppsSqrt_16s_ISfs	1328
7.110.2.2 nppsSqrt_16s_Sfs	1328
7.110.2.3 nppsSqrt_16sc_ISfs	1329
7.110.2.4 nppsSqrt_16sc_Sfs	1329
7.110.2.5 nppsSqrt_16u_ISfs	1329
7.110.2.6 nppsSqrt_16u_Sfs	1330
7.110.2.7 nppsSqrt_32f	1330
7.110.2.8 nppsSqrt_32f_I	1330
7.110.2.9 nppsSqrt_32fc	1330
7.110.2.10 nppsSqrt_32fc_I	1331
7.110.2.11 nppsSqrt_32s16s_Sfs	1331
7.110.2.12 nppsSqrt_64f	1331
7.110.2.13 nppsSqrt_64f_I	1332
7.110.2.14 nppsSqrt_64fc	1332
7.110.2.15 nppsSqrt_64fc_I	1332

7.110.2.16	nppsSqrt_64s16s_Sfs	1332
7.110.2.17	nppsSqrt_64s_ISfs	1333
7.110.2.18	nppsSqrt_64s_Sfs	1333
7.110.2.19	nppsSqrt_8u_ISfs	1333
7.110.2.20	nppsSqrt_8u_Sfs	1333
7.111	Cubrt	1335
7.111.1	Detailed Description	1335
7.111.2	Function Documentation	1335
7.111.2.1	nppsCubrt_32f	1335
7.111.2.2	nppsCubrt_32s16s_Sfs	1335
7.112	Exp	1336
7.112.1	Detailed Description	1336
7.112.2	Function Documentation	1336
7.112.2.1	nppsExp_16s_ISfs	1336
7.112.2.2	nppsExp_16s_Sfs	1337
7.112.2.3	nppsExp_32f	1337
7.112.2.4	nppsExp_32f64f	1337
7.112.2.5	nppsExp_32f_I	1338
7.112.2.6	nppsExp_32s_ISfs	1338
7.112.2.7	nppsExp_32s_Sfs	1338
7.112.2.8	nppsExp_64f	1338
7.112.2.9	nppsExp_64f_I	1339
7.112.2.10	nppsExp_64s_ISfs	1339
7.112.2.11	nppsExp_64s_Sfs	1339
7.113	Ln	1340
7.113.1	Detailed Description	1340
7.113.2	Function Documentation	1340
7.113.2.1	nppsLn_16s_ISfs	1340
7.113.2.2	nppsLn_16s_Sfs	1341
7.113.2.3	nppsLn_32f	1341
7.113.2.4	nppsLn_32f_I	1341
7.113.2.5	nppsLn_32s16s_Sfs	1342
7.113.2.6	nppsLn_32s_ISfs	1342
7.113.2.7	nppsLn_32s_Sfs	1342
7.113.2.8	nppsLn_64f	1343
7.113.2.9	nppsLn_64f32f	1343

7.113.2.10 nppsLn_64f_I	1343
7.114 log10	1344
7.114.1 Detailed Description	1344
7.114.2 Function Documentation	1344
7.114.2.1 npps10Log10_32s_ISfs	1344
7.114.2.2 npps10Log10_32s_Sfs	1344
7.115 SumLn	1345
7.115.1 Detailed Description	1345
7.115.2 Function Documentation	1345
7.115.2.1 nppsSumLn_16s32f	1345
7.115.2.2 nppsSumLn_32f	1346
7.115.2.3 nppsSumLn_32f64f	1346
7.115.2.4 nppsSumLn_64f	1346
7.115.2.5 nppsSumLnGetBufferSize_16s32f	1347
7.115.2.6 nppsSumLnGetBufferSize_32f	1347
7.115.2.7 nppsSumLnGetBufferSize_32f64f	1347
7.115.2.8 nppsSumLnGetBufferSize_64f	1348
7.116 Arctan	1349
7.116.1 Detailed Description	1349
7.116.2 Function Documentation	1349
7.116.2.1 nppsArctan_32f	1349
7.116.2.2 nppsArctan_32f_I	1349
7.116.2.3 nppsArctan_64f	1350
7.116.2.4 nppsArctan_64f_I	1350
7.117 Normalize	1351
7.117.1 Detailed Description	1351
7.117.2 Function Documentation	1351
7.117.2.1 nppsNormalize_16s_Sfs	1351
7.117.2.2 nppsNormalize_16sc_Sfs	1352
7.117.2.3 nppsNormalize_32f	1352
7.117.2.4 nppsNormalize_32fc	1352
7.117.2.5 nppsNormalize_64f	1353
7.117.2.6 nppsNormalize_64fc	1353
7.118 Cauchy, CauchyD, and CauchyDD2	1354
7.118.1 Detailed Description	1354
7.118.2 Function Documentation	1354

7.118.2.1 nppsCauchy_32f_I . . . . .	1354
7.118.2.2 nppsCauchyD_32f_I . . . . .	1354
7.118.2.3 nppsCauchyDD2_32f_I . . . . .	1355
7.119 Logical And Shift Operations . . . . .	1356
7.120 AndC . . . . .	1357
7.120.1 Detailed Description . . . . .	1357
7.120.2 Function Documentation . . . . .	1357
7.120.2.1 nppsAndC_16u . . . . .	1357
7.120.2.2 nppsAndC_16u_I . . . . .	1358
7.120.2.3 nppsAndC_32u . . . . .	1358
7.120.2.4 nppsAndC_32u_I . . . . .	1358
7.120.2.5 nppsAndC_8u . . . . .	1358
7.120.2.6 nppsAndC_8u_I . . . . .	1359
7.121 And . . . . .	1360
7.121.1 Detailed Description . . . . .	1360
7.121.2 Function Documentation . . . . .	1360
7.121.2.1 nppsAnd_16u . . . . .	1360
7.121.2.2 nppsAnd_16u_I . . . . .	1361
7.121.2.3 nppsAnd_32u . . . . .	1361
7.121.2.4 nppsAnd_32u_I . . . . .	1361
7.121.2.5 nppsAnd_8u . . . . .	1361
7.121.2.6 nppsAnd_8u_I . . . . .	1362
7.122 OrC . . . . .	1363
7.122.1 Detailed Description . . . . .	1363
7.122.2 Function Documentation . . . . .	1363
7.122.2.1 nppsOrC_16u . . . . .	1363
7.122.2.2 nppsOrC_16u_I . . . . .	1364
7.122.2.3 nppsOrC_32u . . . . .	1364
7.122.2.4 nppsOrC_32u_I . . . . .	1364
7.122.2.5 nppsOrC_8u . . . . .	1364
7.122.2.6 nppsOrC_8u_I . . . . .	1365
7.123 Or . . . . .	1366
7.123.1 Detailed Description . . . . .	1366
7.123.2 Function Documentation . . . . .	1366
7.123.2.1 nppsOr_16u . . . . .	1366
7.123.2.2 nppsOr_16u_I . . . . .	1367

7.123.2.3	nppsOr_32u	1367
7.123.2.4	nppsOr_32u_I	1367
7.123.2.5	nppsOr_8u	1367
7.123.2.6	nppsOr_8u_I	1368
7.124	XorC	1369
7.124.1	Detailed Description	1369
7.124.2	Function Documentation	1369
7.124.2.1	nppsXorC_16u	1369
7.124.2.2	nppsXorC_16u_I	1370
7.124.2.3	nppsXorC_32u	1370
7.124.2.4	nppsXorC_32u_I	1370
7.124.2.5	nppsXorC_8u	1370
7.124.2.6	nppsXorC_8u_I	1371
7.125	Xor	1372
7.125.1	Detailed Description	1372
7.125.2	Function Documentation	1372
7.125.2.1	nppsXor_16u	1372
7.125.2.2	nppsXor_16u_I	1373
7.125.2.3	nppsXor_32u	1373
7.125.2.4	nppsXor_32u_I	1373
7.125.2.5	nppsXor_8u	1373
7.125.2.6	nppsXor_8u_I	1374
7.126	Not	1375
7.126.1	Detailed Description	1375
7.126.2	Function Documentation	1375
7.126.2.1	nppsNot_16u	1375
7.126.2.2	nppsNot_16u_I	1376
7.126.2.3	nppsNot_32u	1376
7.126.2.4	nppsNot_32u_I	1376
7.126.2.5	nppsNot_8u	1376
7.126.2.6	nppsNot_8u_I	1377
7.127	LShiftC	1378
7.127.1	Detailed Description	1378
7.127.2	Function Documentation	1378
7.127.2.1	nppsLShiftC_16s	1378
7.127.2.2	nppsLShiftC_16s_I	1379

7.127.2.3	nppsLShiftC_16u	1379
7.127.2.4	nppsLShiftC_16u_I	1379
7.127.2.5	nppsLShiftC_32s	1380
7.127.2.6	nppsLShiftC_32s_I	1380
7.127.2.7	nppsLShiftC_32u	1380
7.127.2.8	nppsLShiftC_32u_I	1381
7.127.2.9	nppsLShiftC_8u	1381
7.127.2.10	nppsLShiftC_8u_I	1381
7.128	RShiftC	1382
7.128.1	Detailed Description	1382
7.128.2	Function Documentation	1382
7.128.2.1	nppsRShiftC_16s	1382
7.128.2.2	nppsRShiftC_16s_I	1383
7.128.2.3	nppsRShiftC_16u	1383
7.128.2.4	nppsRShiftC_16u_I	1383
7.128.2.5	nppsRShiftC_32s	1384
7.128.2.6	nppsRShiftC_32s_I	1384
7.128.2.7	nppsRShiftC_32u	1384
7.128.2.8	nppsRShiftC_32u_I	1385
7.128.2.9	nppsRShiftC_8u	1385
7.128.2.10	nppsRShiftC_8u_I	1385
7.129	Conversion Functions	1386
7.130	Convert	1387
7.130.1	Function Documentation	1389
7.130.1.1	nppsConvert_16s32f	1389
7.130.1.2	nppsConvert_16s32f_Sfs	1389
7.130.1.3	nppsConvert_16s32s	1389
7.130.1.4	nppsConvert_16s64f_Sfs	1389
7.130.1.5	nppsConvert_16s8s_Sfs	1389
7.130.1.6	nppsConvert_16u32f	1389
7.130.1.7	nppsConvert_32f16s_Sfs	1389
7.130.1.8	nppsConvert_32f16u_Sfs	1389
7.130.1.9	nppsConvert_32f32s_Sfs	1389
7.130.1.10	nppsConvert_32f64f	1389
7.130.1.11	nppsConvert_32f8s_Sfs	1389
7.130.1.12	nppsConvert_32f8u_Sfs	1389

7.130.1.13	nppsConvert_32s16s . . . . .	1389
7.130.1.14	nppsConvert_32s16s_Sfs . . . . .	1389
7.130.1.15	nppsConvert_32s32f . . . . .	1389
7.130.1.16	nppsConvert_32s32f_Sfs . . . . .	1389
7.130.1.17	nppsConvert_32s64f . . . . .	1389
7.130.1.18	nppsConvert_32s64f_Sfs . . . . .	1389
7.130.1.19	nppsConvert_64f16s_Sfs . . . . .	1389
7.130.1.20	nppsConvert_64f32f . . . . .	1389
7.130.1.21	nppsConvert_64f32s_Sfs . . . . .	1389
7.130.1.22	nppsConvert_64f64s_Sfs . . . . .	1389
7.130.1.23	nppsConvert_64s32s_Sfs . . . . .	1389
7.130.1.24	nppsConvert_64s64f . . . . .	1389
7.130.1.25	nppsConvert_8s16s . . . . .	1389
7.130.1.26	nppsConvert_8s32f . . . . .	1389
7.130.1.27	nppsConvert_8u32f . . . . .	1389
7.131	Threshold . . . . .	1390
7.131.1	Function Documentation . . . . .	1394
7.131.1.1	nppsThreshold_16s . . . . .	1394
7.131.1.2	nppsThreshold_16s_I . . . . .	1395
7.131.1.3	nppsThreshold_16sc . . . . .	1395
7.131.1.4	nppsThreshold_16sc_I . . . . .	1395
7.131.1.5	nppsThreshold_32f . . . . .	1396
7.131.1.6	nppsThreshold_32f_I . . . . .	1396
7.131.1.7	nppsThreshold_32fc . . . . .	1396
7.131.1.8	nppsThreshold_32fc_I . . . . .	1397
7.131.1.9	nppsThreshold_64f . . . . .	1397
7.131.1.10	nppsThreshold_64f_I . . . . .	1398
7.131.1.11	nppsThreshold_64fc . . . . .	1398
7.131.1.12	nppsThreshold_64fc_I . . . . .	1398
7.131.1.13	nppsThreshold_GT_16s . . . . .	1399
7.131.1.14	nppsThreshold_GT_16s_I . . . . .	1399
7.131.1.15	nppsThreshold_GT_16sc . . . . .	1399
7.131.1.16	nppsThreshold_GT_16sc_I . . . . .	1400
7.131.1.17	nppsThreshold_GT_32f . . . . .	1400
7.131.1.18	nppsThreshold_GT_32f_I . . . . .	1400
7.131.1.19	nppsThreshold_GT_32fc . . . . .	1401

---

7.131.1.20	nppsThreshold_GT_32fc_I	1401
7.131.1.21	nppsThreshold_GT_64f	1401
7.131.1.22	nppsThreshold_GT_64f_I	1402
7.131.1.23	nppsThreshold_GT_64fc	1402
7.131.1.24	nppsThreshold_GT_64fc_I	1402
7.131.1.25	nppsThreshold_GTVAl_16s	1403
7.131.1.26	nppsThreshold_GTVAl_16s_I	1403
7.131.1.27	nppsThreshold_GTVAl_16sc	1403
7.131.1.28	nppsThreshold_GTVAl_16sc_I	1404
7.131.1.29	nppsThreshold_GTVAl_32f	1404
7.131.1.30	nppsThreshold_GTVAl_32f_I	1404
7.131.1.31	nppsThreshold_GTVAl_32fc	1405
7.131.1.32	nppsThreshold_GTVAl_32fc_I	1405
7.131.1.33	nppsThreshold_GTVAl_64f	1405
7.131.1.34	nppsThreshold_GTVAl_64f_I	1406
7.131.1.35	nppsThreshold_GTVAl_64fc	1406
7.131.1.36	nppsThreshold_GTVAl_64fc_I	1406
7.131.1.37	nppsThreshold_LT_16s	1407
7.131.1.38	nppsThreshold_LT_16s_I	1407
7.131.1.39	nppsThreshold_LT_16sc	1407
7.131.1.40	nppsThreshold_LT_16sc_I	1408
7.131.1.41	nppsThreshold_LT_32f	1408
7.131.1.42	nppsThreshold_LT_32f_I	1408
7.131.1.43	nppsThreshold_LT_32fc	1409
7.131.1.44	nppsThreshold_LT_32fc_I	1409
7.131.1.45	nppsThreshold_LT_64f	1409
7.131.1.46	nppsThreshold_LT_64f_I	1410
7.131.1.47	nppsThreshold_LT_64fc	1410
7.131.1.48	nppsThreshold_LT_64fc_I	1410
7.131.1.49	nppsThreshold_LTVAl_16s	1411
7.131.1.50	nppsThreshold_LTVAl_16s_I	1411
7.131.1.51	nppsThreshold_LTVAl_16sc	1411
7.131.1.52	nppsThreshold_LTVAl_16sc_I	1412
7.131.1.53	nppsThreshold_LTVAl_32f	1412
7.131.1.54	nppsThreshold_LTVAl_32f_I	1412
7.131.1.55	nppsThreshold_LTVAl_32fc	1413

7.131.1.56	nppsThreshold_LTVal_32fc_I . . . . .	1413
7.131.1.57	nppsThreshold_LTVal_64f . . . . .	1413
7.131.1.58	nppsThreshold_LTVal_64f_I . . . . .	1414
7.131.1.59	nppsThreshold_LTVal_64fc . . . . .	1414
7.131.1.60	nppsThreshold_LTVal_64fc_I . . . . .	1414
7.132	Filtering Functions . . . . .	1415
7.132.1	Detailed Description . . . . .	1415
7.132.2	Function Documentation . . . . .	1415
7.132.2.1	nppsIntegral_32s . . . . .	1415
7.132.2.2	nppsIntegralGetBufferSize_32s . . . . .	1415
7.133	Initialization . . . . .	1416
7.134	Set . . . . .	1417
7.134.1	Function Documentation . . . . .	1417
7.134.1.1	nppsSet_16s . . . . .	1417
7.134.1.2	nppsSet_16sc . . . . .	1418
7.134.1.3	nppsSet_32f . . . . .	1418
7.134.1.4	nppsSet_32fc . . . . .	1418
7.134.1.5	nppsSet_32s . . . . .	1419
7.134.1.6	nppsSet_32sc . . . . .	1419
7.134.1.7	nppsSet_64f . . . . .	1419
7.134.1.8	nppsSet_64fc . . . . .	1419
7.134.1.9	nppsSet_64s . . . . .	1420
7.134.1.10	nppsSet_64sc . . . . .	1420
7.134.1.11	lnppsSet_8u . . . . .	1420
7.135	Zero . . . . .	1421
7.135.1	Function Documentation . . . . .	1421
7.135.1.1	nppsZero_16s . . . . .	1421
7.135.1.2	nppsZero_16sc . . . . .	1422
7.135.1.3	nppsZero_32f . . . . .	1422
7.135.1.4	nppsZero_32fc . . . . .	1422
7.135.1.5	nppsZero_32s . . . . .	1422
7.135.1.6	nppsZero_32sc . . . . .	1423
7.135.1.7	nppsZero_64f . . . . .	1423
7.135.1.8	nppsZero_64fc . . . . .	1423
7.135.1.9	nppsZero_64s . . . . .	1423
7.135.1.10	nppsZero_64sc . . . . .	1424

7.135.1.1 lnppsZero_8u . . . . .	1424
7.136 Copy . . . . .	1425
7.136.1 Function Documentation . . . . .	1425
7.136.1.1 nppsCopy_16s . . . . .	1425
7.136.1.2 nppsCopy_16sc . . . . .	1426
7.136.1.3 nppsCopy_32f . . . . .	1426
7.136.1.4 nppsCopy_32fc . . . . .	1426
7.136.1.5 nppsCopy_32s . . . . .	1427
7.136.1.6 nppsCopy_32sc . . . . .	1427
7.136.1.7 nppsCopy_64fc . . . . .	1427
7.136.1.8 nppsCopy_64s . . . . .	1427
7.136.1.9 nppsCopy_64sc . . . . .	1428
7.136.1.10 nppsCopy_8u . . . . .	1428
7.137 Statistical Functions . . . . .	1429
7.137.1 Detailed Description . . . . .	1429
7.138 MinEvery And MaxEvery Functions . . . . .	1430
7.138.1 Detailed Description . . . . .	1430
7.138.2 Function Documentation . . . . .	1430
7.138.2.1 nppsMaxEvery_16s_I . . . . .	1430
7.138.2.2 nppsMaxEvery_16u_I . . . . .	1431
7.138.2.3 nppsMaxEvery_32f_I . . . . .	1431
7.138.2.4 nppsMaxEvery_32s_I . . . . .	1431
7.138.2.5 nppsMaxEvery_8u_I . . . . .	1432
7.138.2.6 nppsMinEvery_16s_I . . . . .	1432
7.138.2.7 nppsMinEvery_16u_I . . . . .	1432
7.138.2.8 nppsMinEvery_32f_I . . . . .	1432
7.138.2.9 nppsMinEvery_32s_I . . . . .	1433
7.138.2.10 nppsMinEvery_64f_I . . . . .	1433
7.138.2.11 nppsMinEvery_8u_I . . . . .	1433
7.139 Sum . . . . .	1434
7.139.1 Function Documentation . . . . .	1435
7.139.1.1 nppsSum_16s32s_Sfs . . . . .	1435
7.139.1.2 nppsSum_16s_Sfs . . . . .	1435
7.139.1.3 nppsSum_16sc32sc_Sfs . . . . .	1436
7.139.1.4 nppsSum_16sc_Sfs . . . . .	1436
7.139.1.5 nppsSum_32f . . . . .	1436

7.139.1.6	nppsSum_32fc	1437
7.139.1.7	nppsSum_32s_Sfs	1437
7.139.1.8	nppsSum_64f	1438
7.139.1.9	nppsSum_64fc	1438
7.139.1.10	nppsSumGetBufferSize_16s32s_Sfs	1438
7.139.1.11	nppsSumGetBufferSize_16s_Sfs	1439
7.139.1.12	nppsSumGetBufferSize_16sc32sc_Sfs	1439
7.139.1.13	nppsSumGetBufferSize_16sc_Sfs	1439
7.139.1.14	nppsSumGetBufferSize_32f	1439
7.139.1.15	nppsSumGetBufferSize_32fc	1440
7.139.1.16	nppsSumGetBufferSize_32s_Sfs	1440
7.139.1.17	nppsSumGetBufferSize_64f	1440
7.139.1.18	nppsSumGetBufferSize_64fc	1440
7.140	Maximum	1441
7.140.1	Function Documentation	1442
7.140.1.1	nppsMax_16s	1442
7.140.1.2	nppsMax_32f	1443
7.140.1.3	nppsMax_32s	1443
7.140.1.4	nppsMax_64f	1443
7.140.1.5	nppsMaxAbs_16s	1444
7.140.1.6	nppsMaxAbs_32s	1444
7.140.1.7	nppsMaxAbsGetBufferSize_16s	1444
7.140.1.8	nppsMaxAbsGetBufferSize_32s	1445
7.140.1.9	nppsMaxAbsIndx_16s	1445
7.140.1.10	nppsMaxAbsIndx_32s	1445
7.140.1.11	nppsMaxAbsIndxGetBufferSize_16s	1446
7.140.1.12	nppsMaxAbsIndxGetBufferSize_32s	1446
7.140.1.13	nppsMaxGetBufferSize_16s	1446
7.140.1.14	nppsMaxGetBufferSize_32f	1447
7.140.1.15	nppsMaxGetBufferSize_32s	1447
7.140.1.16	nppsMaxGetBufferSize_64f	1447
7.140.1.17	nppsMaxIndx_16s	1447
7.140.1.18	nppsMaxIndx_32f	1448
7.140.1.19	nppsMaxIndx_32s	1448
7.140.1.20	nppsMaxIndx_64f	1449
7.140.1.21	nppsMaxIndxGetBufferSize_16s	1449

7.140.1.22	nppsMaxIndxGetBufferSize_32f	1449
7.140.1.23	nppsMaxIndxGetBufferSize_32s	1450
7.140.1.24	nppsMaxIndxGetBufferSize_64f	1450
7.141	Minimum	1451
7.141.1	Function Documentation	1452
7.141.1.1	nppsMin_16s	1452
7.141.1.2	nppsMin_32f	1453
7.141.1.3	nppsMin_32s	1453
7.141.1.4	nppsMin_64f	1453
7.141.1.5	nppsMinAbs_16s	1454
7.141.1.6	nppsMinAbs_32s	1454
7.141.1.7	nppsMinAbsGetBufferSize_16s	1454
7.141.1.8	nppsMinAbsGetBufferSize_32s	1455
7.141.1.9	nppsMinAbsIndx_16s	1455
7.141.1.10	nppsMinAbsIndx_32s	1455
7.141.1.11	nppsMinAbsIndxGetBufferSize_16s	1456
7.141.1.12	nppsMinAbsIndxGetBufferSize_32s	1456
7.141.1.13	nppsMinGetBufferSize_16s	1456
7.141.1.14	nppsMinGetBufferSize_32f	1457
7.141.1.15	nppsMinGetBufferSize_32s	1457
7.141.1.16	nppsMinGetBufferSize_64f	1457
7.141.1.17	nppsMinIndx_16s	1457
7.141.1.18	nppsMinIndx_32f	1458
7.141.1.19	nppsMinIndx_32s	1458
7.141.1.20	nppsMinIndx_64f	1459
7.141.1.21	nppsMinIndxGetBufferSize_16s	1459
7.141.1.22	nppsMinIndxGetBufferSize_32f	1459
7.141.1.23	nppsMinIndxGetBufferSize_32s	1460
7.141.1.24	nppsMinIndxGetBufferSize_64f	1460
7.142	Mean	1461
7.142.1	Function Documentation	1462
7.142.1.1	nppsMean_16s_Sfs	1462
7.142.1.2	nppsMean_16sc_Sfs	1462
7.142.1.3	nppsMean_32f	1462
7.142.1.4	nppsMean_32fc	1463
7.142.1.5	nppsMean_32s_Sfs	1463

---

7.142.1.6	nppsMean_64f	1464
7.142.1.7	nppsMean_64fc	1464
7.142.1.8	nppsMeanGetBufferSize_16s_Sfs	1464
7.142.1.9	nppsMeanGetBufferSize_16sc_Sfs	1465
7.142.1.10	nppsMeanGetBufferSize_32f	1465
7.142.1.11	nppsMeanGetBufferSize_32fc	1465
7.142.1.12	nppsMeanGetBufferSize_32s_Sfs	1465
7.142.1.13	nppsMeanGetBufferSize_64f	1466
7.142.1.14	nppsMeanGetBufferSize_64fc	1466
7.143	Standard Deviation	1467
7.143.1	Function Documentation	1467
7.143.1.1	nppsStdDev_16s32s_Sfs	1467
7.143.1.2	nppsStdDev_16s_Sfs	1468
7.143.1.3	nppsStdDev_32f	1468
7.143.1.4	nppsStdDev_64f	1468
7.143.1.5	nppsStdDevGetBufferSize_16s32s_Sfs	1469
7.143.1.6	nppsStdDevGetBufferSize_16s_Sfs	1469
7.143.1.7	nppsStdDevGetBufferSize_32f	1469
7.143.1.8	nppsStdDevGetBufferSize_64f	1469
7.144	Mean And Standard Deviation	1470
7.144.1	Function Documentation	1470
7.144.1.1	nppsMeanStdDev_16s32s_Sfs	1470
7.144.1.2	nppsMeanStdDev_16s_Sfs	1471
7.144.1.3	nppsMeanStdDev_32f	1471
7.144.1.4	nppsMeanStdDev_64f	1471
7.144.1.5	nppsMeanStdDevGetBufferSize_16s32s_Sfs	1472
7.144.1.6	nppsMeanStdDevGetBufferSize_16s_Sfs	1472
7.144.1.7	nppsMeanStdDevGetBufferSize_32f	1472
7.144.1.8	nppsMeanStdDevGetBufferSize_64f	1473
7.145	Minimum_Maximum	1474
7.145.1	Function Documentation	1476
7.145.1.1	nppsMinMax_16s	1476
7.145.1.2	nppsMinMax_16u	1476
7.145.1.3	nppsMinMax_32f	1476
7.145.1.4	nppsMinMax_32s	1477
7.145.1.5	nppsMinMax_32u	1477

7.145.1.6	nppsMinMax_64f	1477
7.145.1.7	nppsMinMax_8u	1478
7.145.1.8	nppsMinMaxGetBufferSize_16s	1478
7.145.1.9	nppsMinMaxGetBufferSize_16u	1478
7.145.1.10	nppsMinMaxGetBufferSize_32f	1479
7.145.1.11	nppsMinMaxGetBufferSize_32s	1479
7.145.1.12	nppsMinMaxGetBufferSize_32u	1479
7.145.1.13	nppsMinMaxGetBufferSize_64f	1480
7.145.1.14	nppsMinMaxGetBufferSize_8u	1480
7.145.1.15	nppsMinMaxIndx_16s	1480
7.145.1.16	nppsMinMaxIndx_16u	1481
7.145.1.17	nppsMinMaxIndx_32f	1481
7.145.1.18	nppsMinMaxIndx_32s	1481
7.145.1.19	nppsMinMaxIndx_32u	1482
7.145.1.20	nppsMinMaxIndx_64f	1482
7.145.1.21	nppsMinMaxIndx_8u	1483
7.145.1.22	nppsMinMaxIndxGetBufferSize_16s	1483
7.145.1.23	nppsMinMaxIndxGetBufferSize_16u	1483
7.145.1.24	nppsMinMaxIndxGetBufferSize_32f	1484
7.145.1.25	nppsMinMaxIndxGetBufferSize_32s	1484
7.145.1.26	nppsMinMaxIndxGetBufferSize_32u	1484
7.145.1.27	nppsMinMaxIndxGetBufferSize_64f	1484
7.145.1.28	nppsMinMaxIndxGetBufferSize_8u	1485
7.146	Infinity Norm	1486
7.146.1	Function Documentation	1487
7.146.1.1	nppsNorm_Inf_16s32f	1487
7.146.1.2	nppsNorm_Inf_16s32s_Sfs	1487
7.146.1.3	nppsNorm_Inf_32f	1487
7.146.1.4	nppsNorm_Inf_32fc32f	1488
7.146.1.5	nppsNorm_Inf_64f	1488
7.146.1.6	nppsNorm_Inf_64fc64f	1488
7.146.1.7	nppsNormInfGetBufferSize_16s32f	1489
7.146.1.8	nppsNormInfGetBufferSize_16s32s_Sfs	1489
7.146.1.9	nppsNormInfGetBufferSize_32f	1489
7.146.1.10	nppsNormInfGetBufferSize_32fc32f	1489
7.146.1.11	nppsNormInfGetBufferSize_64f	1490

7.146.1.12	nppsNormInfGetBufferSize_64fc64f . . . . .	1490
7.147	L1 Norm . . . . .	1491
7.147.1	Function Documentation . . . . .	1492
7.147.1.1	nppsNorm_L1_16s32f . . . . .	1492
7.147.1.2	nppsNorm_L1_16s32s_Sfs . . . . .	1492
7.147.1.3	nppsNorm_L1_16s64s_Sfs . . . . .	1492
7.147.1.4	nppsNorm_L1_32f . . . . .	1493
7.147.1.5	nppsNorm_L1_32fc64f . . . . .	1493
7.147.1.6	nppsNorm_L1_64f . . . . .	1493
7.147.1.7	nppsNorm_L1_64fc64f . . . . .	1494
7.147.1.8	nppsNormL1GetBufferSize_16s32f . . . . .	1494
7.147.1.9	nppsNormL1GetBufferSize_16s32s_Sfs . . . . .	1494
7.147.1.10	nppsNormL1GetBufferSize_16s64s_Sfs . . . . .	1495
7.147.1.11	nppsNormL1GetBufferSize_32f . . . . .	1495
7.147.1.12	nppsNormL1GetBufferSize_32fc64f . . . . .	1495
7.147.1.13	nppsNormL1GetBufferSize_64f . . . . .	1495
7.147.1.14	nppsNormL1GetBufferSize_64fc64f . . . . .	1496
7.148	L2 Norm . . . . .	1497
7.148.1	Function Documentation . . . . .	1498
7.148.1.1	nppsNorm_L2_16s32f . . . . .	1498
7.148.1.2	nppsNorm_L2_16s32s_Sfs . . . . .	1498
7.148.1.3	nppsNorm_L2_32f . . . . .	1498
7.148.1.4	nppsNorm_L2_32fc64f . . . . .	1499
7.148.1.5	nppsNorm_L2_64f . . . . .	1499
7.148.1.6	nppsNorm_L2_64fc64f . . . . .	1499
7.148.1.7	nppsNorm_L2Sqr_16s64s_Sfs . . . . .	1500
7.148.1.8	nppsNormL2GetBufferSize_16s32f . . . . .	1500
7.148.1.9	nppsNormL2GetBufferSize_16s32s_Sfs . . . . .	1500
7.148.1.10	nppsNormL2GetBufferSize_32f . . . . .	1501
7.148.1.11	nppsNormL2GetBufferSize_32fc64f . . . . .	1501
7.148.1.12	nppsNormL2GetBufferSize_64f . . . . .	1501
7.148.1.13	nppsNormL2GetBufferSize_64fc64f . . . . .	1501
7.148.1.14	nppsNormL2SqrGetBufferSize_16s64s_Sfs . . . . .	1502
7.149	Infinity Norm Diff . . . . .	1503
7.149.1	Function Documentation . . . . .	1504
7.149.1.1	nppsNormDiff_Inf_16s32f . . . . .	1504

7.149.1.2	nppsNormDiff_Inf_16s32s_Sfs	1504
7.149.1.3	nppsNormDiff_Inf_32f	1504
7.149.1.4	nppsNormDiff_Inf_32fc32f	1505
7.149.1.5	nppsNormDiff_Inf_64f	1505
7.149.1.6	nppsNormDiff_Inf_64fc64f	1506
7.149.1.7	nppsNormDiffInfGetBufferSize_16s32f	1506
7.149.1.8	nppsNormDiffInfGetBufferSize_16s32s_Sfs	1506
7.149.1.9	nppsNormDiffInfGetBufferSize_32f	1506
7.149.1.10	nppsNormDiffInfGetBufferSize_32fc32f	1507
7.149.1.11	nppsNormDiffInfGetBufferSize_64f	1507
7.149.1.12	nppsNormDiffInfGetBufferSize_64fc64f	1507
7.150L1	Norm Diff	1508
7.150.1	Function Documentation	1509
7.150.1.1	nppsNormDiff_L1_16s32f	1509
7.150.1.2	nppsNormDiff_L1_16s32s_Sfs	1509
7.150.1.3	nppsNormDiff_L1_16s64s_Sfs	1509
7.150.1.4	nppsNormDiff_L1_32f	1510
7.150.1.5	nppsNormDiff_L1_32fc64f	1510
7.150.1.6	nppsNormDiff_L1_64f	1511
7.150.1.7	nppsNormDiff_L1_64fc64f	1511
7.150.1.8	nppsNormDiffL1GetBufferSize_16s32f	1511
7.150.1.9	nppsNormDiffL1GetBufferSize_16s32s_Sfs	1512
7.150.1.10	nppsNormDiffL1GetBufferSize_16s64s_Sfs	1512
7.150.1.11	nppsNormDiffL1GetBufferSize_32f	1512
7.150.1.12	nppsNormDiffL1GetBufferSize_32fc64f	1512
7.150.1.13	nppsNormDiffL1GetBufferSize_64f	1513
7.150.1.14	nppsNormDiffL1GetBufferSize_64fc64f	1513
7.151L2	Norm Diff	1514
7.151.1	Function Documentation	1515
7.151.1.1	nppsNormDiff_L2_16s32f	1515
7.151.1.2	nppsNormDiff_L2_16s32s_Sfs	1515
7.151.1.3	nppsNormDiff_L2_32f	1515
7.151.1.4	nppsNormDiff_L2_32fc64f	1516
7.151.1.5	nppsNormDiff_L2_64f	1516
7.151.1.6	nppsNormDiff_L2_64fc64f	1517
7.151.1.7	nppsNormDiff_L2Sqr_16s64s_Sfs	1517

7.151.1.8 nppsNormDiffL2GetBufferSize_16s32f	1517
7.151.1.9 nppsNormDiffL2GetBufferSize_16s32s_Sfs	1518
7.151.1.10 nppsNormDiffL2GetBufferSize_32f	1518
7.151.1.11 nppsNormDiffL2GetBufferSize_32fc64f	1518
7.151.1.12 nppsNormDiffL2GetBufferSize_64f	1518
7.151.1.13 nppsNormDiffL2GetBufferSize_64fc64f	1519
7.151.1.14 nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs	1519
7.152 Dot Product	1520
7.152.1 Function Documentation	1523
7.152.1.1 nppsDotProd_16s16sc32fc	1523
7.152.1.2 nppsDotProd_16s16sc32sc_Sfs	1524
7.152.1.3 nppsDotProd_16s16sc64sc	1524
7.152.1.4 nppsDotProd_16s16sc_Sfs	1525
7.152.1.5 nppsDotProd_16s32f	1525
7.152.1.6 nppsDotProd_16s32s32s_Sfs	1525
7.152.1.7 nppsDotProd_16s32s_Sfs	1526
7.152.1.8 nppsDotProd_16s64s	1526
7.152.1.9 nppsDotProd_16s_Sfs	1527
7.152.1.10 nppsDotProd_16sc32fc	1527
7.152.1.11 nppsDotProd_16sc32sc_Sfs	1527
7.152.1.12 nppsDotProd_16sc64sc	1528
7.152.1.13 nppsDotProd_16sc_Sfs	1528
7.152.1.14 nppsDotProd_32f	1529
7.152.1.15 nppsDotProd_32f32fc	1529
7.152.1.16 nppsDotProd_32f32fc64fc	1529
7.152.1.17 nppsDotProd_32f64f	1530
7.152.1.18 nppsDotProd_32fc	1530
7.152.1.19 nppsDotProd_32fc64fc	1530
7.152.1.20 nppsDotProd_32s32sc_Sfs	1531
7.152.1.21 nppsDotProd_32s_Sfs	1531
7.152.1.22 nppsDotProd_32sc_Sfs	1531
7.152.1.23 nppsDotProd_64f	1532
7.152.1.24 nppsDotProd_64f64fc	1532
7.152.1.25 nppsDotProd_64fc	1533
7.152.1.26 nppsDotProdGetBufferSize_16s16sc32fc	1533
7.152.1.27 nppsDotProdGetBufferSize_16s16sc32sc_Sfs	1533

7.152.1.28	nppsDotProdGetBufferSize_16s16sc64sc	1533
7.152.1.29	nppsDotProdGetBufferSize_16s16sc_Sfs	1534
7.152.1.30	nppsDotProdGetBufferSize_16s32f	1534
7.152.1.31	nppsDotProdGetBufferSize_16s32s32s_Sfs	1534
7.152.1.32	nppsDotProdGetBufferSize_16s32s_Sfs	1535
7.152.1.33	nppsDotProdGetBufferSize_16s64s	1535
7.152.1.34	nppsDotProdGetBufferSize_16s_Sfs	1535
7.152.1.35	nppsDotProdGetBufferSize_16sc32fc	1535
7.152.1.36	nppsDotProdGetBufferSize_16sc32sc_Sfs	1536
7.152.1.37	nppsDotProdGetBufferSize_16sc64sc	1536
7.152.1.38	nppsDotProdGetBufferSize_16sc_Sfs	1536
7.152.1.39	nppsDotProdGetBufferSize_32f	1536
7.152.1.40	nppsDotProdGetBufferSize_32f32fc	1537
7.152.1.41	nppsDotProdGetBufferSize_32f32fc64fc	1537
7.152.1.42	nppsDotProdGetBufferSize_32f64f	1537
7.152.1.43	nppsDotProdGetBufferSize_32fc	1537
7.152.1.44	nppsDotProdGetBufferSize_32fc64fc	1538
7.152.1.45	nppsDotProdGetBufferSize_32s32sc_Sfs	1538
7.152.1.46	nppsDotProdGetBufferSize_32s_Sfs	1538
7.152.1.47	nppsDotProdGetBufferSize_32sc_Sfs	1538
7.152.1.48	nppsDotProdGetBufferSize_64f	1539
7.152.1.49	nppsDotProdGetBufferSize_64f64fc	1539
7.152.1.50	nppsDotProdGetBufferSize_64fc	1539
7.153	Count In Range	1540
7.153.1	Function Documentation	1540
7.153.1.1	nppsCountInRange_32s	1540
7.153.1.2	nppsCountInRangeGetBufferSize_32s	1540
7.154	Count Zero Crossings	1541
7.154.1	Function Documentation	1541
7.154.1.1	nppsZeroCrossing_16s32f	1541
7.154.1.2	nppsZeroCrossing_32f	1541
7.154.1.3	nppsZeroCrossingGetBufferSize_16s32f	1542
7.154.1.4	nppsZeroCrossingGetBufferSize_32f	1542
7.155	Memory Management	1543
<b>8</b>	<b>Data Structure Documentation</b>	<b>1545</b>
8.1	Npp16sc Struct Reference	1545

---

8.1.1	Detailed Description	1545
8.1.2	Field Documentation	1545
8.1.2.1	im	1545
8.1.2.2	re	1545
8.2	Npp16uc Struct Reference	1546
8.2.1	Detailed Description	1546
8.2.2	Field Documentation	1546
8.2.2.1	im	1546
8.2.2.2	re	1546
8.3	Npp32fc Struct Reference	1547
8.3.1	Detailed Description	1547
8.3.2	Field Documentation	1547
8.3.2.1	im	1547
8.3.2.2	re	1547
8.4	Npp32sc Struct Reference	1548
8.4.1	Detailed Description	1548
8.4.2	Field Documentation	1548
8.4.2.1	im	1548
8.4.2.2	re	1548
8.5	Npp32uc Struct Reference	1549
8.5.1	Detailed Description	1549
8.5.2	Field Documentation	1549
8.5.2.1	im	1549
8.5.2.2	re	1549
8.6	Npp64fc Struct Reference	1550
8.6.1	Detailed Description	1550
8.6.2	Field Documentation	1550
8.6.2.1	im	1550
8.6.2.2	re	1550
8.7	Npp64sc Struct Reference	1551
8.7.1	Detailed Description	1551
8.7.2	Field Documentation	1551
8.7.2.1	im	1551
8.7.2.2	re	1551
8.8	Npp8uc Struct Reference	1552
8.8.1	Detailed Description	1552

8.8.2	Field Documentation	1552
8.8.2.1	im	1552
8.8.2.2	re	1552
8.9	NppiHaarBuffer Struct Reference	1553
8.9.1	Field Documentation	1553
8.9.1.1	haarBuffer	1553
8.9.1.2	haarBufferSize	1553
8.10	NppiHaarClassifier_32f Struct Reference	1554
8.10.1	Field Documentation	1554
8.10.1.1	classifiers	1554
8.10.1.2	classifierSize	1554
8.10.1.3	classifierStep	1554
8.10.1.4	counterDevice	1554
8.10.1.5	numClassifiers	1554
8.11	NppiPoint Struct Reference	1555
8.11.1	Detailed Description	1555
8.11.2	Field Documentation	1555
8.11.2.1	x	1555
8.11.2.2	y	1555
8.12	NppiRect Struct Reference	1556
8.12.1	Detailed Description	1556
8.12.2	Field Documentation	1556
8.12.2.1	height	1556
8.12.2.2	width	1556
8.12.2.3	x	1556
8.12.2.4	y	1556
8.13	NppiSize Struct Reference	1557
8.13.1	Detailed Description	1557
8.13.2	Field Documentation	1557
8.13.2.1	height	1557
8.13.2.2	width	1557
8.14	NppLibraryVersion Struct Reference	1558
8.14.1	Field Documentation	1558
8.14.1.1	build	1558
8.14.1.2	major	1558
8.14.1.3	minor	1558

# Chapter 1

## NVIDIA Performance Primitives

IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE  
As of NPP version 5.0 and beyond a few parameters for a few pre-5.0 existing image LUT functions have changed from host memory pointers to device memory pointers. Your application will fail (crash or report an error) if you use these functions with host memory pointers. The functions are the `nppiLUT_Linear_8u_xxx` functions.

Also, pre-5.0 function `nppiMeanStdDev8uC1RGetBufferHostSize` has been renamed `nppiMeanStdDevGetBufferHostSize_8u_C1R`.

### 1.1 What is NPP?

NVIDIA NPP is a library of functions for performing CUDA accelerated processing. The initial set of functionality in the library focuses on imaging and video processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

NPP can be used in one of two ways:

- A stand-alone library for adding GPU acceleration to an application with minimal effort. Using this route allows developers to add GPU acceleration to their applications in a matter of hours.
- A cooperative library for interoperating with a developer's GPU code efficiently.

Either route allows developers to harness the massive compute resources of NVIDIA GPUs, while simultaneously reducing development times.

### 1.2 Documentation

- [General API Conventions](#)
- [Signal-Processing Specific API Conventions](#)
- [Imaging-Processing Specific API Conventions](#)

## 1.3 Technical Specifications

Supported Platforms:

- Microsoft Windows 7 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Microsoft Windows XP (64-bit and 32-bit)
- Linux (Centos & Ubuntu) (64-bit and 32-bit)
- Mac OS X

## 1.4 Files

NPP is comprises the following files:

### 1.4.1 Header Files

- [nppdefs.h](#)
- [nppcore.h](#)
- [nppi.h](#)
- [npps.h](#)
- [nppversion.h](#)
- [npp.h](#)

All those header files are located in the CUDA Toolkit's

```
/include/
```

directory.

### 1.4.2 Library Files

On the Windows platform the NPP stub library is found in the CUDA Toolkit's library directory:

```
/lib/npp.lib
```

The matching DLL is located in the CUDA Toolkit's binary directory:

```
/bin/npp32_50_9.dll // Dynamic library for 32-bit Windows.  
/bin/npp64_50_9.dll // Dynamic library for 64-bit Windows.
```

On Linux and Mac platforms the dynamic libraries are located in the lib directory

```
/lib/libnpp32.so.5.0.9 // NPP 32-bit dynamic library for Linux  
/lib/libnpp64.so.5.0.9 // NPP 64-bit dynamic library for Linux  
  
/lib/libnpp32.5.0.dylib // NPP 32-bit dynamic library for Mac  
/lib/libnpp64.5.0.dylib // NPP 64-bit dynamic library for Mac
```

## 1.5 Supported NVIDIA Hardware

NPP runs on all CUDA capable NVIDIA hardware. For details please see [http://www.nvidia.com/object/cuda\\_learn\\_products.html](http://www.nvidia.com/object/cuda_learn_products.html)



## **Chapter 2**

# **General API Conventions**

## 2.1 Memory Management

The design of all the NPP functions follows the same guidelines as other NVIDIA CUDA libraries like cuFFT and cuBLAS. That is that all pointer arguments in those APIs are device pointers.

This convention enables the individual developer to make smart choices about memory management that minimize the number of memory transfers. It also allows the user the maximum flexibility regarding which of the various memory transfer mechanisms offered by the CUDA runtime is used, e.g. synchronous or asynchronous memory transfers, zero-copy and pinned memory, etc.

The most basic steps involved in using NPP for processing data is as follows:

1. Transfer input data from the host to device using

```
cudaMemcpy(...)
```

2. Process data using one or several NPP functions or custom CUDA kernels
3. Transfer the result data from the device to the host using

```
cudaMemcpy(...)
```

### 2.1.1 Scratch Buffer and Host Pointer

Some primitives of NPP require additional device memory buffers (scratch buffers) for calculations, e.g. signal and image reductions (Sum, Max, Min, MinMax, etc.). In order to give the NPP user maximum control regarding memory allocations and performance, it is the user's responsibility to allocate and delete those temporary buffers. For one this has the benefit that the library will not allocate memory unbeknownst to the user. It also allows developers who invoke the same primitive repeatedly to allocate the scratch only once, improving performance and potential device-memory fragmentation.

Scratch-buffer memory is unstructured and may be passed to the primitive in uninitialized form. This allows for reuse of the same scratch buffers with any primitive require scratch memory, as long as it is sufficiently sized.

The minimum scratch-buffer size for a given primitive (e.g. [nppsSum\\_32f\(\)](#)) can be obtained by a companion function (e.g. [nppsSumGetBufferSize\\_32f\(\)](#)). The buffer size is returned via a host pointer as allocation of the scratch-buffer is performed via CUDA runtime host code.

An example to invoke Sum primitive and allocate and free the necessary scratch memory:

```
...
// Compute the appropriate size of the scratch-memory buffer
int nBufferSize;
nppsSumGetBufferSize_32f(nLength, &nBufferSize);
// Allocate the scratch buffer
Npp8u * pDeviceBuffer;
cudaMalloc((void **)&pDeviceBuffer, nBufferSize);
// Call the primitive with the scratch buffer
nppsSum_32f(pSrc, nLength, pSum, nppAlgHintNone, pDeviceBuffer);
// Free the scratch buffer
cudaFree(pDeviceBuffer);
...
```

## 2.2 Function Naming

Since NPP is a C API and therefore does not allow for function overloading for different data-types the NPP naming convention addresses the need to differentiate between different flavors of the same algorithm

or primitive function but for various data types. This disambiguation of different flavors of a primitive is done via a suffix containing data type and other disambiguating information.

In addition to the flavor suffix, all NPP functions are prefixed with by the letters "npp". Primitives belonging to NPP's image-processing module add the letter "i" to the npp prefix, i.e. are prefixed by "nppi". Similarly signal-processing primitives are prefixed with "npps".

The general naming scheme is:

npp<module info><PrimitiveName>\_<data-type info>[\_<additional flavor info>](<parameter list>)

The data-type information uses the same names as the [Basic NPP Data Types](#). For example the data-type information "8u" would imply that the primitive operates on [Npp8u](#) data.

If a primitive consumes different type data from what it produces, both types will be listed in the order of consumed to produced data type.

Details about the "additional flavor information" is provided for each of the NPP modules, since each problem domain uses different flavor information suffixes.

## 2.3 Integer Result Scaling

NPP signal processing and imaging primitives often operate on integer data. This integer data is usually a fixed point fractional representation of some physical magnitue (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend to produce results exceeding the original fixed-point range if treated as regular integers.

In cases where the results exceed the original range, these functions clamp the result values back to the valid range. E.g. the maximum positive value for a 16-bit unsigned integer is 32767. A multiplication operation of  $4 * 10000 = 40000$  would exceed this range. The result would be clamped to be 32767.

To avoid the level of lost information due to clamping most integer primitives allow for result scaling. Primitives with result scaling have the "Sfs" suffix in their name and provide a parameter "nScaleFactor" that controls the amount of scaling. Before the results of an operation are clamped to the valid output-data range by multiplying them with  $2^{-nScaleFactor}$ .

Example: The primitive [nppsSqr\\_8u\\_Sfs\(\)](#) computes the square of 8-bit unsigned sample values in a signal (1D array of values). The maximum value of a 8-bit value is 255. The square of  $255^2 = 65025$  which would be clamped to 255 if no result scaling is performed. In order to map the maximum value of 255 to 255 in the result, one would specify an integer result scaling factor of 8, i.e. multiply each result with  $2^{-8} = \frac{1}{2^8} = \frac{1}{256}$ . The final result for a signal value of 255 being squared and scaled would be:

$$255^2 \cdot 2^{-8} = 254.00390625$$

which would be rounded to a final result of 254.

A medium gray value of 128 would result in

$$128^2 * 2^{-8} = 64$$



## **Chapter 3**

# **Signal-Processing Specific API Conventions**

## 3.1 Signal Data

Signal data is passed to and from NPPS primitives via a pointer to the signal's data type.

The general idea behind this fairly low-level way of passing signal data is ease-of-adoption into existing software projects:

- Passing the data pointer rather than a higher-level signal struct allows for easy adoption by not requiring a specific signal representation (that could include total signal size offset, or other additional information). This avoids awkward packing and unpacking of signal data from the host application to an NPP specific signal representation.

### 3.1.1 Parameter Names for Signal Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

Those are signals consumed by the algorithm.

#### 3.1.1.1 Source Signal Pointer

The source signal data is generally passed via a pointer named

```
pSrc
```

The source signal pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppsPrimitive_32s(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

#### 3.1.1.2 Destination Signal Pointer

The destination signal data is generally passed via a pointer named

```
pDst
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

```
pDst1, pDst2, ...
```

#### 3.1.1.3 In-Place Signal Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place signal data are called:

```
pSrcDst
```

### 3.1.2 Signal Data Alignment Requirements

NPP requires signal sample data to be naturally aligned, i.e. any pointer

```
NppType * p;
```

to a sample in a signal needs to fulfill:

```
assert(p % sizeof(p) == 0);
```

### 3.1.3 Signal Data Related Error Codes

All NPPI primitives operating on signal data validate the signal-data pointer for proper alignment and test that the point is not null.

Failed validation results in one of the following error codes being returned and the primitive not being executed:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if the image-data pointer is 0 (NULL).
- [NPP\\_ALIGNMENT\\_ERROR](#) if the signal-data pointer address is not a multiple of the signal's data-type size.

## 3.2 Signal Length

The vast majority of NPPS functions take a

```
nLength
```

parameter that tells the primitive how many of the signal's samples starting from the given data pointer are to be processed.

### 3.2.1 Length Related Error Codes

All NPPS primitives taking a length parameter validate this input.

Failed validation results in the following error code being returned and the primitive not being executed:

- [NPP\\_SIZE\\_ERROR](#) is returned if the length is negative.



## **Chapter 4**

# **Imaging-Processing Specific API Conventions**

## 4.1 Function Naming

Image processing related functions use a number of suffixes to indicate various different flavors of a primitive beyond just different data types. The flavor suffix uses the following abbreviations:

- "A" if the image is a 4 channel image this indicates the result alpha channel is not affected by the primitive.
- "Cn" the image consists of n channel packed pixels, where n can be 1, 2, 3 or 4.
- "Pn" the image consists of n separate image planes, where n can be 1, 2, 3 or 4.
- "C" (following the channel information) indicates that the primitive only operates on one of the color channels, the "channel-of-interest". All other output channels are not affected by the primitive.
- "I" indicates that the primitive works "in-place". In this case the image-data pointer is usually named "pSrcDst" to indicate that the image data serves as source and destination at the same time.
- "M" indicates "masked operation". These types of primitives have an additional "mask image" as input. Each pixel in the destination image corresponds to a pixel in the mask image. Only pixels with a corresponding non-zero mask pixel are being processed.
- "R" indicates the primitive operates only on a rectangular "region-of-interest" or "ROI". All ROI primitives take an additional input parameter of type [NppiSize](#), which specifies the width and height of the rectangular region that the primitive should process. For details on how primitives operate on ROIs see: [Region-of-Interest \(ROI\)](#).
- "Sfs" indicates the result values are processed by fixed scaling and saturation before they're written out.

The suffixes above always appear in alphabetical order. E.g. a 4 channel primitive not affecting the alpha channel with masked operation, in place and with scaling/saturation and ROI would have the postfix: "AC4IMRSfs".

## 4.2 Image Data

Image data is passed to and from NPPI primitives via a pair of parameters:

1. A pointer to the image's underlying data type.
2. A line step in bytes (also sometimes called line stride).

The general idea behind this fairly low-level way of passing image data is ease-of-adoption into existing software projects:

- Passing a raw pointer to the underlying pixel data type, rather than structured (by color) channel pixel data allows usage of the function in a wide variety of situations avoiding risky type cast or expensive image data copies.
- Passing the data pointer and line step individually rather than a higher-level image struct again allows for easy adoption by not requiring a specific image representation and thus avoiding awkward packing and unpacking of image data from the host application to an NPP specific image representation.

### 4.2.1 Line Step

The line step (also called "line stride" or "row step") allows lines of oddly sized images to start on well-aligned addresses by adding a number of unused bytes at the ends of the lines. This type of line padding has been common practice in digital image processing for a long time and is not particular to GPU image processing.

The line step is the number of bytes in a line **including the padding**. An other way to interpret this number is to say that it is the number of bytes between the first pixel of successive rows in the image, or generally the number of bytes between two neighboring pixels in any column of pixels.

The general reason for the existence of the line step it is that uniformly aligned rows of pixel enable optimizations of memory-access patterns.

Even though all functions in NPP will work with arbitrarily aligned images, best performance can only be achieved with well aligned image data. Any image data allocated with the NPP image allocators or the 2D memory allocators in the CUDA runtime, is well aligned.

Particularly on older CUDA capable GPUs it is likely that the performance decrease for misaligned data is substantial (orders of magnitude).

All image data passed to NPPI primitives requires a line step to be provided. It is important to keep in mind that this line step is always specified in terms of bytes, not pixels.

### 4.2.2 Parameter Names for Image Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

#### 4.2.2.1 Passing Source-Image Data

Those are images consumed by the algorithm.

##### 4.2.2.1.1 Source-Image Pointer

The source image data is generally passed via a pointer named

```
pSrc
```

The source image pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppiPrimitive_32s_C1R(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

##### 4.2.2.1.2 Source-Planar-Image Pointer Array

The planar source image data is generally passed via an array of pointers named

```
pSrc[]
```

The planar source image pointer array is generally defined a constant array of constant pointers, enforcing that the primitive does not change any image data pointed to by those pointers. E.g.

```
nppiPrimitive_8u_P3R(const Npp8u * const pSrc[3], ...)
```

Each pointer in the array points to a different image plane.

#### 4.2.2.1.3 Source-Planar-Image Pointer

The multiple plane source image data is passed via a set of pointers named

```
pSrc1, pSrc2, ...
```

The planar source image pointer is generally defined as one of a set of constant pointers with each pointer pointing to a different input image plane.

#### 4.2.2.1.4 Source-Image Line Step

The source image line step is the number of bytes between successive rows in the image. The source image line step parameter is

```
nSrcStep
```

or in the case of multiple source images

```
nSrcStep1, nSrcStep2, ...
```

#### 4.2.2.1.5 Source-Planar-Image Line Step Array

The source planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the input image. The source planar image line step array parameter is

```
rSrcStep[]
```

#### 4.2.2.1.6 Source-Planar-Image Line Step

The source planar image line step is the number of bytes between successive rows in a particular plane of the multiplane input image. The source planar image line step parameter is

```
nSrcStep1, nSrcStep2, ...
```

#### 4.2.2.2 Passing Destination-Image Data

Those are images produced by the algorithm.

#### 4.2.2.2.1 Destination-Image Pointer

The destination image data is generally passed via a pointer named

```
pDst
```

In case the primitive generates multiple images as outputs the destination pointers are numbered like this:

```
pDst1, pDst2, ...
```

#### 4.2.2.2.2 Destination-Planar-Image Pointer Array

The planar destination image data pointers are generally passed via an array of pointers named

```
pDst[]
```

Each pointer in the array points to a different image plane.

#### 4.2.2.2.3 Destination-Planar-Image Pointer

The destination planar image data is generally passed via a pointer to each plane of a multiplane output image named

```
pDst1, pDst2, ...
```

#### 4.2.2.2.4 Destination-Image Line Step

The destination image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

```
nDstStep1, nDstStep2, ...
```

#### 4.2.2.2.5 Destination-Planar-Image Line Step Array

The destination planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the output image. The destination planar image line step array parameter is

```
rDstStep[]
```

#### 4.2.2.2.6 Destination-Planar-Image Line Step

The destination planar image line step is the number of bytes between successive rows for a particular plane in a multiplane output image. The destination planar image line step parameter is

```
nDstStep1, nDstStep2, ...
```

### 4.2.2.3 Passing In-Place Image Data

#### 4.2.2.3.1 In-Place Image Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place image data are called:

```
pSrcDst
```

#### 4.2.2.3.2 In-Place-Image Line Step

The in-place line step parameter is

```
nSrcDstStep
```

### 4.2.2.4 Passing Mask-Image Data

Some image processing primitives have variants supporting [Masked Operation](#).

#### 4.2.2.4.1 Mask-Image Pointer

The mask-image data is generally passed via a pointer named

```
pMask
```

#### 4.2.2.4.2 Mask-Image Line Step

The mask-image line step parameter is

```
nMaskStep
```

### 4.2.2.5 Passing Channel-of-Interest Data

Some image processing primitives support [Channel-of-Interest API](#).

#### 4.2.2.5.1 Channel\_of\_Interest Number

The channel-of-interest data is generally an integer (either 1, 2, or 3):

```
nCOI
```

## 4.2.3 Image Data Alignment Requirements

NPP requires pixel data to adhere to certain alignment constraints: For 2 and 4 channel images the following alignment requirement holds: `data_pointer % (#channels * sizeof(channel type)) == 0`. E.g. a 4 channel image with underlying type `Npp8u` (8-bit unsigned) would require all pixels to fall on addresses that are multiples of 4 (4 channels \* 1 byte size).

As a logical consequence of all pixels being aligned to their natural size the image line steps of 2 and 4 channel images also need to be multiples of the pixel size.

1 and 3 channel images only require that pixel pointers are aligned to the underlying data type, i.e. `pData % sizeof(data type) == 0`. And consequentially line steps are also held to this requirement.

#### 4.2.4 Image Data Related Error Codes

All NPPI primitives operating on image data validate the image-data pointer for proper alignment and test that the point is not null. They also validate the line stride for proper alignment and guard against the step being less or equal to 0. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_STEP_ERROR` is returned if the data step is 0 or negative.
- `NPP_NOT_EVEN_STEP_ERROR` is returned if the line step is not a multiple of the pixel size for 2 and 4 channel images.
- `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL).
- `NPP_ALIGNMENT_ERROR` if the image-data pointer address is not a multiple of the pixel size for 2 and 4 channel images.

### 4.3 Region-of-Interest (ROI)

In practice processing a rectangular sub-region of an image is often more common than processing complete images. The vast majority of NPP's image-processing primitives allow for processing of such sub regions also referred to as regions-of-interest or ROIs.

All primitives supporting ROI processing are marked by a "R" in their name suffix. Where possible, the ROI a primitive operates on is passed as a single `NppiSize` struct, which provides the with and height of the ROI. This raises the obvious question how the primitive knows where in the image this rectangle of (width, height) is located. The "start pixel" of the ROI is implicitly given by the image-data pointer. I.e. instead of explicitly passing a pixel coordinate for the upper-right corner of the ROI the primitive's user needs to perform the necessary offset computation on the image data pointers, such that the pointers passed to the primitive thus point to the start of the ROI.

In practice this means that for an image (`pSrc`, `nSrcStep`) and the start-pixel of the ROI being given by (`xROI`, `yROI`), one would pass

```
pSrcOffset = pSrc + yROI * nSrcStep + xROI * PixelSize;
```

as the image-data source to the primitive. `PixelSize` is typically computed as

```
PixelSize = NumberOfColorChannels * sizeof(PixelDataType).
```

E.g. for a primitive like `nppiSet_16s_C4R()` we would have

- `NumberOfColorChannels == 4;`
- `sizeof(Npp16s) == 2;`
- and thus `PixelSize = 4 * 2 = 8;`

### 4.3.1 ROI Related Error Codes

All NPPI primitives operating on ROIs of image data validate the ROI size and image's step size. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_SIZE_ERROR` is returned if either the ROI width or ROI height are negative.
- `NPP_STEP_ERROR` is returned if the ROI width exceeds the image's line step. In mathematical terms  $(\text{widthROI} * \text{PixelSize}) > \text{nLinStep}$  indicates an error.

## 4.4 Masked Operation

Some primitive support masked operation. An "M" in the suffix of those variants indicates masked operation. Primitives supporting masked operation consume an additional input image provided via a [Mask-Image Pointer](#) and [Mask-Image Line Step](#). The mask image is interpreted by these primitives as a boolean image. The values of type `Npp8u` are interpreted as boolean values where a values of 0 indicates false, any non-zero values true.

Unless otherwise indicated the operation is only performed on pixels where its spatially corresponding mask pixel is true (non-zero). E.g. a masked copy operation would only copy those pixels in the ROI that have corresponding non-zero mask pixels.

## 4.5 Channel-of-Interest API

Some primitives allow restricting operations to a single channel of interest within a multi-channel image. These primitives are suffixed with the letter "C" (after the channel information, e.g. `nppiCopy_8u_C3CR(...)`). The channel-of-interest is generally selected by offsetting the image-data pointer to point directly to the channel- of-interest rather than the base of the first pixel in the ROI. Some primitives also explicitly specify the selected channel number and pass it via an integer, e.g. `nppiMean_StdDev_8u_C3CR(...)`.

### 4.5.1 Select-Channel Source-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the source image. E.g. if `pSrc` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy the second channel of this source image into the first channel of a destination image given by `pDst` by offsetting the pointer by two data items:

```
nppiCopy_8u_C3CR(pSrc + 2, nSrcStep, pDst, nDstStep, oSizeROI);
```

### 4.5.2 Select-Channel Source-Image

Some primitives allow the user to select the channel-of-interest by specifying the channel number (`nCOI`). This approach is typically used in the image statistical functions. For example,

```
nppiMean_StdDev_8u_C3CR(pSrc, nSrcStep, oSizeROI, nCOI, pDeviceBuffer, pMean, pStdDev );
```

The channel-of-interest number can be either 1, 2, or 3.

### 4.5.3 Select-Channel Destination-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the destination image. E.g. if pDst is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy data into the second channel of this destination image from the first channel of a source image given by pSrc by offsetting the destination pointer by two data items:

```
nppiCopy_8u_C3CR(pSrc, nSrcStep, pDst + 2, nDstStep, oSizeROI);
```



# Chapter 5

## Module Index

### 5.1 Modules

Here is a list of all modules:

NPP Core	29
NPP Type Definitions and Constants	32
Basic NPP Data Types	42
NPP Image Processing	45
Arithmetic and Logical Operations	46
Arithmetic Operations	47
AddC	49
MulC	75
MulCScale	101
SubC	108
DivC	134
AbsDiffC	160
Add	162
AddSquare	191
AddProduct	194
AddWeighted	198
Mul	202
MulScale	231
Sub	240
Div	270
Div_Round	299
Abs	314
AbsDiff	321
Sqr	324
Sqrt	338
Ln	350
Exp	357
Logical Operations	364
AndC	365
OrC	376
XorC	387
RShiftC	398
LShiftC	415

And	426
Or	438
Xor	450
Not	462
Alpha Composition	466
AlphaCompC	467
AlphaPremulC	475
AlphaComp	482
AlphaPremul	489
Color and Sampling Conversion	491
Color Model Conversion	492
Color Sampling Format Conversion	560
Color Gamma Correction	588
Complement Color Key	594
Color Processing	597
Compression	602
Quantization Functions	603
Labeling and Segmentation	606
GraphCut	607
Data Exchange and Initialization	614
Set	615
Copy	640
Convert	684
Copy Constant Border	725
Transpose And Swap Channels	728
Filtering Functions	730
1D Linear Filter	731
1D Window Sum	734
Convolution	736
2D Fixed Linear Filters	738
Rank Filters	740
Geometry Transforms	743
Resize	744
Rotate	746
Mirror	755
Affine Transforms	762
Perspective Transform	812
Linear Transforms	858
Fourier Transforms	859
Morphological Operations	861
Dilation And Erosion	862
Statistics Functions	865
Sum	866
Minimum	880
Maximum	907
Minimum_Maximum	934
Mean	965
Mean And Standard Deviation	986
Infinity Norm	1001
L1 Norm	1022
L2 Norm	1042
Norm Diff	1062
Integral and Rectangular Standard Deviation	1064

Histogram . . . . .	1066
Memory Management . . . . .	1093
Threshold and Compare Operations . . . . .	1105
Threshold Operations . . . . .	1106
Compare Operations . . . . .	1195
NPP Signal Processing . . . . .	1217
Arithmetic and Logical Operations . . . . .	1218
Arithmetic Operations . . . . .	1219
AddC . . . . .	1221
AddProductC . . . . .	1230
MulC . . . . .	1231
SubC . . . . .	1241
SubCRev . . . . .	1250
DivC . . . . .	1259
DivCRev . . . . .	1266
Add . . . . .	1268
AddProduct . . . . .	1280
Mul . . . . .	1284
Sub . . . . .	1297
Div . . . . .	1307
Div_Round . . . . .	1315
Abs . . . . .	1318
Sqr . . . . .	1321
Sqrt . . . . .	1327
Cubrt . . . . .	1335
Exp . . . . .	1336
Ln . . . . .	1340
10Log10 . . . . .	1344
SumLn . . . . .	1345
Arctan . . . . .	1349
Normalize . . . . .	1351
Cauchy, CauchyD, and CauchyDD2 . . . . .	1354
Logical And Shift Operations . . . . .	1356
AndC . . . . .	1357
And . . . . .	1360
OrC . . . . .	1363
Or . . . . .	1366
XorC . . . . .	1369
Xor . . . . .	1372
Not . . . . .	1375
LShiftC . . . . .	1378
RShiftC . . . . .	1382
Conversion Functions . . . . .	1386
Convert . . . . .	1387
Threshold . . . . .	1390
Filtering Functions . . . . .	1415
Initialization . . . . .	1416
Set . . . . .	1417
Zero . . . . .	1421
Copy . . . . .	1425
Statistical Functions . . . . .	1429
MinEvery And MaxEvery Functions . . . . .	1430
Sum . . . . .	1434

---

Maximum . . . . .	1441
Minimum . . . . .	1451
Mean . . . . .	1461
Standard Deviation . . . . .	1467
Mean And Standard Deviation . . . . .	1470
Minimum_Maximum . . . . .	1474
Infinity Norm . . . . .	1486
L1 Norm . . . . .	1491
L2 Norm . . . . .	1497
Infinity Norm Diff . . . . .	1503
L1 Norm Diff . . . . .	1508
L2 Norm Diff . . . . .	1514
Dot Product . . . . .	1520
Count In Range . . . . .	1540
Count Zero Crossings . . . . .	1541
Memory Management . . . . .	1543

# Chapter 6

## Data Structure Index

### 6.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">Npp16sc</a> (Complex Number This struct represents a short complex number ) . . . . .	1545
<a href="#">Npp16uc</a> (Complex Number This struct represents an unsigned short complex number ) . . . . .	1546
<a href="#">Npp32fc</a> (Complex Number This struct represents a single floating-point complex number ) . . . . .	1547
<a href="#">Npp32sc</a> (Complex Number This struct represents a signed int complex number ) . . . . .	1548
<a href="#">Npp32uc</a> (Complex Number This struct represents an unsigned int complex number ) . . . . .	1549
<a href="#">Npp64fc</a> (Complex Number This struct represents a double floating-point complex number ) . . . . .	1550
<a href="#">Npp64sc</a> (Complex Number This struct represents a long long complex number ) . . . . .	1551
<a href="#">Npp8uc</a> (Complex Number This struct represents an unsigned char complex number ) . . . . .	1552
<a href="#">NppiHaarBuffer</a> . . . . .	1553
<a href="#">NppiHaarClassifier_32f</a> . . . . .	1554
<a href="#">NppiPoint</a> (2D Point ) . . . . .	1555
<a href="#">NppiRect</a> (2D Rectangle This struct contains position and size information of a rectangle in two space ) . . . . .	1556
<a href="#">NppiSize</a> (2D Size This struct typically represents the size of a a rectangular region in two space ) . . . . .	1557
<a href="#">NppLibraryVersion</a> . . . . .	1558



# Chapter 7

## Module Documentation

### 7.1 NPP Core

Basic functions for library management, in particular library version and device property query functions.

#### Functions

- const [NppLibraryVersion](#) \* [nppGetLibVersion](#) (void)  
*Get the NPP library version.*
- [NppGpuComputeCapability](#) [nppGetGpuComputeCapability](#) (void)  
*What CUDA compute model is supported by the active CUDA device?*
- int [nppGetGpuNumSMs](#) (void)  
*Get the number of Streaming Multiprocessors (SM) on the active CUDA device.*
- int [nppGetMaxThreadsPerBlock](#) (void)  
*Get the maximum number of threads per block on the active CUDA device.*
- int [nppGetMaxThreadsPerSM](#) (void)  
*Get the maximum number of threads per SM for the active GPU.*
- const char \* [nppGetGpuName](#) (void)  
*Get the name of the active CUDA device.*
- [cudaStream\\_t](#) [nppGetStream](#) (void)  
*Get the NPP CUDA stream.*
- void [nppSetStream](#) ([cudaStream\\_t](#) hStream)  
*Set the NPP CUDA stream.*

#### 7.1.1 Detailed Description

Basic functions for library management, in particular library version and device property query functions.

## 7.1.2 Function Documentation

### 7.1.2.1 `NppGpuComputeCapability nppGetGpuComputeCapability (void)`

What CUDA compute model is supported by the active CUDA device?

Before trying to call any NPP functions, the user should make a call this function to ensure that the current machine has a CUDA capable device.

**Returns:**

An enum value representing if a CUDA capable device was found and what level of compute capabilities it supports.

### 7.1.2.2 `const char* nppGetGpuName (void)`

Get the name of the active CUDA device.

**Returns:**

Name string of the active graphics-card/compute device in a system.

### 7.1.2.3 `int nppGetGpuNumSMs (void)`

Get the number of Streaming Multiprocessors (SM) on the active CUDA device.

**Returns:**

Number of SMs of the default CUDA device.

### 7.1.2.4 `const NppLibraryVersion* nppGetLibVersion (void)`

Get the NPP library version.

**Returns:**

A struct containing separate values for major and minor revision and build number.

### 7.1.2.5 `int nppGetMaxThreadsPerBlock (void)`

Get the maximum number of threads per block on the active CUDA device.

**Returns:**

Maximum number of threads per block on the active CUDA device.

**7.1.2.6 int nppGetMaxThreadsPerSM (void)**

Get the maximum number of threads per SM for the active GPU.

**Returns:**

Maximum number of threads per SM for the active GPU

**7.1.2.7 cudaStream\_t nppGetStream (void)**

Get the NPP CUDA stream.

NPP enables concurrent device tasks via a global stream state variable. The NPP stream by default is set to stream 0, i.e. non-concurrent mode. A user can set the NPP stream to any valid CUDA stream. All CUDA commands issued by NPP (e.g. kernels launched by the NPP library) are then issued to that NPP stream.

**7.1.2.8 void nppSetStream (cudaStream\_t *hStream*)**

Set the NPP CUDA stream.

**See also:**

[nppGetStream\(\)](#)

## 7.2 NPP Type Definitions and Constants

### Data Structures

- struct [NppLibraryVersion](#)
- struct [NppiPoint](#)  
*2D Point*
- struct [NppiSize](#)  
*2D Size This struct typically represents the size of a rectangular region in two space.*
- struct [NppiRect](#)  
*2D Rectangle This struct contains position and size information of a rectangle in two space.*
- struct [NppiHaarClassifier\\_32f](#)
- struct [NppiHaarBuffer](#)

### Modules

- [Basic NPP Data Types](#)

### Defines

- #define [NPP\\_MIN\\_8U](#) ( 0 )  
*Minimum 8-bit unsigned integer.*
- #define [NPP\\_MAX\\_8U](#) ( 255 )  
*Maximum 8-bit unsigned integer.*
- #define [NPP\\_MIN\\_16U](#) ( 0 )  
*Minimum 16-bit unsigned integer.*
- #define [NPP\\_MAX\\_16U](#) ( 65535 )  
*Maximum 16-bit unsigned integer.*
- #define [NPP\\_MIN\\_32U](#) ( 0 )  
*Minimum 32-bit unsigned integer.*
- #define [NPP\\_MAX\\_32U](#) ( 4294967295U )  
*Maximum 32-bit unsigned integer.*
- #define [NPP\\_MIN\\_64U](#) ( 0 )  
*Minimum 64-bit unsigned integer.*
- #define [NPP\\_MAX\\_64U](#) ( 18446744073709551615ULL )  
*Maximum 64-bit unsigned integer.*
- #define [NPP\\_MIN\\_8S](#) (-127 - 1 )  
*Minimum 8-bit signed integer.*

- #define `NPP_MAX_8S` ( 127 )  
*Maximum 8-bit signed integer.*
- #define `NPP_MIN_16S` (-32767 - 1 )  
*Minimum 16-bit signed integer.*
- #define `NPP_MAX_16S` ( 32767 )  
*Maximum 16-bit signed integer.*
- #define `NPP_MIN_32S` (-2147483647 - 1 )  
*Minimum 32-bit signed integer.*
- #define `NPP_MAX_32S` ( 2147483647 )  
*Maximum 32-bit signed integer.*
- #define `NPP_MAX_64S` ( 9223372036854775807LL )  
*Maximum 64-bit signed integer.*
- #define `NPP_MIN_64S` (-9223372036854775807LL - 1)  
*Minimum 64-bit signed integer.*
- #define `NPP_MINABS_32F` ( 1.175494351e-38f )  
*Smallest positive 32-bit floating point value.*
- #define `NPP_MAXABS_32F` ( 3.402823466e+38f )  
*Largest positive 32-bit floating point value.*
- #define `NPP_MINABS_64F` ( 2.2250738585072014e-308 )  
*Smallest positive 64-bit floating point value.*
- #define `NPP_MAXABS_64F` ( 1.7976931348623158e+308 )  
*Largest positive 64-bit floating point value.*

## Enumerations

- enum `NppiInterpolationMode` {  
`NPPI_INTER_UNDEFINED` = 0,  
`NPPI_INTER_NN` = 1,  
`NPPI_INTER_LINEAR` = 2,  
`NPPI_INTER_CUBIC` = 4,  
`NPPI_INTER_CUBIC2P_BSPLINE`,  
`NPPI_INTER_CUBIC2P_CATMULLROM`,  
`NPPI_INTER_CUBIC2P_B05C03`,  
`NPPI_INTER_SUPER` = 8,  
`NPPI_INTER_LANCZOS` = 16,  
`NPPI_SMOOTH_EDGE` = (1 << 31) }

*Filtering methods.*

- enum NppStatus {  
NPP\_NOT\_SUPPORTED\_MODE\_ERROR = -9999,  
NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ERROR = -213,  
NPP\_RESIZE\_NO\_OPERATION\_ERROR = -50,  
NPP\_COI\_ERROR = -29,  
NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR = -28,  
NPP\_NOT\_SUFFICIENT\_COMPUTE\_CAPABILITY = -27,  
NPP\_BAD\_ARG\_ERROR = -26,  
NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR = -25,  
NPP\_TEXTURE\_BIND\_ERROR = -24,  
NPP\_COEFF\_ERROR = -23,  
NPP\_RECT\_ERROR = -22,  
NPP\_QUAD\_ERROR = -21,  
NPP\_WRONG\_INTERSECTION\_ROI\_ERROR = -20,  
NPP\_NOT\_EVEN\_STEP\_ERROR = -19,  
NPP\_INTERPOLATION\_ERROR = -18,  
NPP\_RESIZE\_FACTOR\_ERROR = -17,  
NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR = -16,  
NPP\_MEMFREE\_ERR = -15,  
NPP\_MEMSET\_ERR = -14,  
NPP\_MEMCPY\_ERROR = -13,  
NPP\_MEM\_ALLOC\_ERR = -12,  
NPP\_HISTO\_NUMBER\_OF\_LEVELS\_ERROR = -11,  
NPP\_MIRROR\_FLIP\_ERR = -10,  
NPP\_INVALID\_INPUT = -9,  
NPP\_ALIGNMENT\_ERROR = -8,  
NPP\_STEP\_ERROR = -7,  
NPP\_SIZE\_ERROR = -6,  
NPP\_POINTER\_ERROR = -5,  
NPP\_NULL\_POINTER\_ERROR = -4,  
NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR = -3,  
NPP\_NOT\_IMPLEMENTED\_ERROR = -2,  
NPP\_ERROR = -1,  
NPP\_NO\_ERROR = 0,  
NPP\_SUCCESS = NPP\_NO\_ERROR,  
NPP\_WARNING = 1,  
NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING = 2,  
NPP\_MISALIGNED\_DST\_ROI\_WARNING = 3,  
NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING = 4,

```
NPP_DOUBLE_SIZE_WARNING = 5,  
NPP_ODD_ROI_WARNING = 6,  
NPP_WRONG_INTERSECTION_ROI_WARNING = 29 }
```

*Error Status Codes.*

- enum NppGpuComputeCapability {  
NPP\_CUDA\_UNKNOWN\_VERSION = -1,  
NPP\_CUDA\_NOT\_CAPABLE,  
NPP\_CUDA\_1\_0,  
NPP\_CUDA\_1\_1,  
NPP\_CUDA\_1\_2,  
NPP\_CUDA\_1\_3,  
NPP\_CUDA\_2\_0,  
NPP\_CUDA\_2\_1,  
NPP\_CUDA\_3\_0 }
- enum NppiAxis {  
NPP\_HORIZONTAL\_AXIS,  
NPP\_VERTICAL\_AXIS,  
NPP\_BOTH\_AXIS }
- enum NppCmpOp {  
NPP\_CMP\_LESS,  
NPP\_CMP\_LESS\_EQ,  
NPP\_CMP\_EQ,  
NPP\_CMP\_GREATER\_EQ,  
NPP\_CMP\_GREATER }
- enum NppRoundMode {  
NPP\_RND\_ZERO,  
NPP\_RND\_NEAR,  
NPP\_RND\_FINANCIAL }
- enum NppiBorderType {  
NPP\_BORDER\_UNDEFINED = -1,  
NPP\_BORDER\_NONE = NPP\_BORDER\_UNDEFINED,  
NPP\_BORDER\_CONSTANT = 0,  
NPP\_BORDER\_REPLICATE = 1,  
NPP\_BORDER\_WRAP = 2 }
- enum NppiAlphaOp {  
NPPI\_OP\_ALPHA\_OVER,  
NPPI\_OP\_ALPHA\_IN,  
NPPI\_OP\_ALPHA\_OUT,  
NPPI\_OP\_ALPHA\_ATOP,  
NPPI\_OP\_ALPHA\_XOR,  
NPPI\_OP\_ALPHA\_PLUS,

```
NPPI_OP_ALPHA_OVER_PREMUL,  
NPPI_OP_ALPHA_IN_PREMUL,  
NPPI_OP_ALPHA_OUT_PREMUL,  
NPPI_OP_ALPHA_ATOP_PREMUL,  
NPPI_OP_ALPHA_XOR_PREMUL,  
NPPI_OP_ALPHA_PLUS_PREMUL,  
NPPI_OP_ALPHA_PREMUL }  
• enum NppsZCType {  
    nppZCR,  
    nppZCXor,  
    nppZCC }
```

## 7.2.1 Define Documentation

### 7.2.1.1 #define NPP\_MAX\_16S ( 32767 )

Maximum 16-bit signed integer.

### 7.2.1.2 #define NPP\_MAX\_16U ( 65535 )

Maximum 16-bit unsigned integer.

### 7.2.1.3 #define NPP\_MAX\_32S ( 2147483647 )

Maximum 32-bit signed integer.

### 7.2.1.4 #define NPP\_MAX\_32U ( 4294967295U )

Maximum 32-bit unsigned integer.

### 7.2.1.5 #define NPP\_MAX\_64S ( 9223372036854775807LL )

Maximum 64-bit signed integer.

### 7.2.1.6 #define NPP\_MAX\_64U ( 18446744073709551615ULL )

Maximum 64-bit unsigned integer.

### 7.2.1.7 #define NPP\_MAX\_8S ( 127 )

Maximum 8-bit signed integer.

### 7.2.1.8 #define NPP\_MAX\_8U ( 255 )

Maximum 8-bit unsigned integer.

**7.2.1.9 #define NPP\_MAXABS\_32F ( 3.402823466e+38f )**

Largest positive 32-bit floating point value.

**7.2.1.10 #define NPP\_MAXABS\_64F ( 1.7976931348623158e+308 )**

Largest positive 64-bit floating point value.

**7.2.1.11 #define NPP\_MIN\_16S (-32767 - 1 )**

Minimum 16-bit signed integer.

**7.2.1.12 #define NPP\_MIN\_16U ( 0 )**

Minimum 16-bit unsigned integer.

**7.2.1.13 #define NPP\_MIN\_32S (-2147483647 - 1 )**

Minimum 32-bit signed integer.

**7.2.1.14 #define NPP\_MIN\_32U ( 0 )**

Minimum 32-bit unsigned integer.

**7.2.1.15 #define NPP\_MIN\_64S (-9223372036854775807LL - 1 )**

Minimum 64-bit signed integer.

**7.2.1.16 #define NPP\_MIN\_64U ( 0 )**

Minimum 64-bit unsigned integer.

**7.2.1.17 #define NPP\_MIN\_8S (-127 - 1 )**

Minimum 8-bit signed integer.

**7.2.1.18 #define NPP\_MIN\_8U ( 0 )**

Minimum 8-bit unsigned integer.

**7.2.1.19 #define NPP\_MINABS\_32F ( 1.175494351e-38f )**

Smallest positive 32-bit floating point value.

### 7.2.1.20 #define NPP\_MINABS\_64F ( 2.2250738585072014e-308 )

Smallest positive 64-bit floating point value.

## 7.2.2 Enumeration Type Documentation

### 7.2.2.1 enum NppCmpOp

Enumerator:

*NPP\_CMP\_LESS*  
*NPP\_CMP\_LESS\_EQ*  
*NPP\_CMP\_EQ*  
*NPP\_CMP\_GREATER\_EQ*  
*NPP\_CMP\_GREATER*

### 7.2.2.2 enum NppGpuComputeCapability

Enumerator:

*NPP\_CUDA\_UNKNOWN\_VERSION* Indicates that the compute-capability query failed.  
*NPP\_CUDA\_NOT\_CAPABLE* Indicates that no CUDA capable device was found.  
*NPP\_CUDA\_1\_0* Indicates that CUDA 1.0 capable device is machine's default device.  
*NPP\_CUDA\_1\_1* Indicates that CUDA 1.1 capable device is machine's default device.  
*NPP\_CUDA\_1\_2* Indicates that CUDA 1.2 capable device is machine's default device.  
*NPP\_CUDA\_1\_3* Indicates that CUDA 1.3 capable device is machine's default device.  
*NPP\_CUDA\_2\_0* Indicates that CUDA 2.0 capable device is machine's default device.  
*NPP\_CUDA\_2\_1* Indicates that CUDA 2.1 capable device is machine's default device.  
*NPP\_CUDA\_3\_0* Indicates that CUDA 3.0 or better is machine's default device.

### 7.2.2.3 enum NppiAlphaOp

Enumerator:

*NPPI\_OP\_ALPHA\_OVER*  
*NPPI\_OP\_ALPHA\_IN*  
*NPPI\_OP\_ALPHA\_OUT*  
*NPPI\_OP\_ALPHA\_ATOP*  
*NPPI\_OP\_ALPHA\_XOR*  
*NPPI\_OP\_ALPHA\_PLUS*  
*NPPI\_OP\_ALPHA\_OVER\_PREMUL*  
*NPPI\_OP\_ALPHA\_IN\_PREMUL*  
*NPPI\_OP\_ALPHA\_OUT\_PREMUL*  
*NPPI\_OP\_ALPHA\_ATOP\_PREMUL*  
*NPPI\_OP\_ALPHA\_XOR\_PREMUL*  
*NPPI\_OP\_ALPHA\_PLUS\_PREMUL*  
*NPPI\_OP\_ALPHA\_PREMUL*

#### 7.2.2.4 enum NppiAxis

Enumerator:

*NPP\_HORIZONTAL\_AXIS*  
*NPP\_VERTICAL\_AXIS*  
*NPP\_BOTH\_AXIS*

#### 7.2.2.5 enum NppiBorderType

Enumerator:

*NPP\_BORDER\_UNDEFINED*  
*NPP\_BORDER\_NONE*  
*NPP\_BORDER\_CONSTANT*  
*NPP\_BORDER\_REPLICATE*  
*NPP\_BORDER\_WRAP*

#### 7.2.2.6 enum NppiInterpolationMode

Filtering methods.

Enumerator:

*NPPI\_INTER\_UNDEFINED*  
*NPPI\_INTER\_NN* Nearest neighbor filtering.  
*NPPI\_INTER\_LINEAR* Linear interpolation.  
*NPPI\_INTER\_CUBIC* Cubic interpolation.  
*NPPI\_INTER\_CUBIC2P\_BSPLINE* Two-parameter cubic filter (B=1, C=0).  
*NPPI\_INTER\_CUBIC2P\_CATMULLROM* Two-parameter cubic filter (B=0, C=1/2).  
*NPPI\_INTER\_CUBIC2P\_B05C03* Two-parameter cubic filter (B=1/2, C=3/10).  
*NPPI\_INTER\_SUPER* Super sampling.  
*NPPI\_INTER\_LANCZOS* Lanczos filtering.  
*NPPI\_SMOOTH\_EDGE* Smooth edge filtering.

#### 7.2.2.7 enum NppRoundMode

Enumerator:

*NPP\_RND\_ZERO*  
*NPP\_RND\_NEAR*  
*NPP\_RND\_FINANCIAL*

### 7.2.2.8 enum NppStatus

Error Status Codes.

Almost all NPP function return error-status information using these return codes. Negative return codes indicate errors, positive return codes indicate warnings, a return code of 0 indicates success.

**Enumerator:**

***NPP\_NOT\_SUPPORTED\_MODE\_ERROR***  
***NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ERROR***  
***NPP\_RESIZE\_NO\_OPERATION\_ERROR***  
***NPP\_COI\_ERROR*** Channel of interest is not 1, 2, or 3.  
***NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR*** ZeroCrossing mode not supported.  
***NPP\_NOT\_SUFFICIENT\_COMPUTE\_CAPABILITY***  
***NPP\_BAD\_ARG\_ERROR***  
***NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR***  
***NPP\_TEXTURE\_BIND\_ERROR***  
***NPP\_COEFF\_ERROR***  
***NPP\_RECT\_ERROR***  
***NPP\_QUAD\_ERROR***  
***NPP\_WRONG\_INTERSECTION\_ROI\_ERROR***  
***NPP\_NOT\_EVEN\_STEP\_ERROR***  
***NPP\_INTERPOLATION\_ERROR***  
***NPP\_RESIZE\_FACTOR\_ERROR***  
***NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR***  
***NPP\_MEMFREE\_ERR***  
***NPP\_MEMSET\_ERR***  
***NPP\_MEMCPY\_ERROR***  
***NPP\_MEM\_ALLOC\_ERR***  
***NPP\_HISTO\_NUMBER\_OF\_LEVELS\_ERROR***  
***NPP\_MIRROR\_FLIP\_ERR***  
***NPP\_INVALID\_INPUT***  
***NPP\_ALIGNMENT\_ERROR***  
***NPP\_STEP\_ERROR*** Step is less or equal zero.  
***NPP\_SIZE\_ERROR***  
***NPP\_POINTER\_ERROR***  
***NPP\_NULL\_POINTER\_ERROR***  
***NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR***  
***NPP\_NOT\_IMPLEMENTED\_ERROR***  
***NPP\_ERROR***  
***NPP\_NO\_ERROR*** Error free operation.  
***NPP\_SUCCESS*** Successful operation (same as NPP\_NO\_ERROR).  
***NPP\_WARNING***

***NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING***

***NPP\_MISALIGNED\_DST\_ROI\_WARNING*** Speed reduction due to uncoalesced memory accesses warning.

***NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING*** Indicates that the quadrangle passed to one of affine warping functions doesn't have necessary properties. First 3 vertices are used, the fourth vertex discarded.

***NPP\_DOUBLE\_SIZE\_WARNING*** Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

***NPP\_ODD\_ROI\_WARNING*** Indicates that for 422/411/420 sampling the ROI width/height was forced to even value.

***NPP\_WRONG\_INTERSECTION\_ROI\_WARNING*** ROI doesn't intersect source or destination ROI/image. No operation performed.

### 7.2.2.9 enum NppsZCType

**Enumerator:**

***nppZCR*** sign change

***nppZCXor*** sign change XOR

***nppZCC*** sign change count\_0

## 7.3 Basic NPP Data Types

### Data Structures

- struct [Npp8uc](#)  
*Complex Number This struct represents an unsigned char complex number.*
- struct [Npp16uc](#)  
*Complex Number This struct represents an unsigned short complex number.*
- struct [Npp16sc](#)  
*Complex Number This struct represents a short complex number.*
- struct [Npp32uc](#)  
*Complex Number This struct represents an unsigned int complex number.*
- struct [Npp32sc](#)  
*Complex Number This struct represents a signed int complex number.*
- struct [Npp32fc](#)  
*Complex Number This struct represents a single floating-point complex number.*
- struct [Npp64sc](#)  
*Complex Number This struct represents a long long complex number.*
- struct [Npp64fc](#)  
*Complex Number This struct represents a double floating-point complex number.*

### Typedefs

- typedef unsigned char [Npp8u](#)  
*8-bit unsigned chars*
- typedef signed char [Npp8s](#)  
*8-bit signed chars*
- typedef unsigned short [Npp16u](#)  
*16-bit unsigned integers*
- typedef short [Npp16s](#)  
*16-bit signed integers*
- typedef unsigned int [Npp32u](#)  
*32-bit unsigned integers*
- typedef int [Npp32s](#)  
*32-bit signed integers*

- typedef unsigned long long [Npp64u](#)  
*64-bit unsigned integers*
- typedef long long [Npp64s](#)  
*64-bit signed integers*
- typedef float [Npp32f](#)  
*32-bit (IEEE) floating-point numbers*
- typedef double [Npp64f](#)  
*64-bit floating-point numbers*

### 7.3.1 Typedef Documentation

#### 7.3.1.1 typedef short [Npp16s](#)

16-bit signed integers

#### 7.3.1.2 typedef unsigned short [Npp16u](#)

16-bit unsigned integers

#### 7.3.1.3 typedef float [Npp32f](#)

32-bit (IEEE) floating-point numbers

#### 7.3.1.4 typedef int [Npp32s](#)

32-bit signed integers

#### 7.3.1.5 typedef unsigned int [Npp32u](#)

32-bit unsigned integers

#### 7.3.1.6 typedef double [Npp64f](#)

64-bit floating-point numbers

#### 7.3.1.7 typedef long long [Npp64s](#)

64-bit signed integers

#### 7.3.1.8 typedef unsigned long long [Npp64u](#)

64-bit unsigned integers

**7.3.1.9 typedef signed char Npp8s**

8-bit signed chars

**7.3.1.10 typedef unsigned char Npp8u**

8-bit unsigned chars

## 7.4 NPP Image Processing

### Modules

- [Arithmetic and Logical Operations](#)
  - [Color and Sampling Conversion](#)  
*Routines manipulating an image's color model and sampling format.*
- [Compression](#)  
*Image compression primitives.*
- [Labeling and Segmentation](#)  
*Pixel labeling and image segmentation operations.*
- [Data Exchange and Initialization](#)  
*Primitives for initialization, copying and converting image data.*
- [Filtering Functions](#)  
*Linear and non-linear image filtering functions.*
- [Geometry Transforms](#)  
*Routines manipulating an image's geometry.*
- [Linear Transforms](#)  
*Linear image transformations.*
- [Morphological Operations](#)  
*Morphological image operations.*
- [Statistics Functions](#)  
*Routines computing statistical image information.*
- [Memory Management](#)  
*Routines for allocating and deallocating pitched image storage.*
- [Threshold and Compare Operations](#)  
*Methods for pixel-wise threshold and compare operations.*

## 7.5 Arithmetic and Logical Operations

### Modules

- [Arithmetic Operations](#)
- [Logical Operations](#)
- [Alpha Composition](#)

## 7.6 Arithmetic Operations

### Modules

- **AddC**  
*Adds a constant value to each pixel of an image.*
- **MulC**  
*Multiplies each pixel of an image by a constant value.*
- **MulCScale**  
*Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.*
- **SubC**  
*Subtracts a constant value from each pixel of an image.*
- **DivC**  
*Divides each pixel of an image by a constant value.*
- **AbsDiffC**  
*Determines absolute difference between each pixel of an image and a constant value.*
- **Add**  
*Pixel by pixel addition of two images.*
- **AddSquare**  
*Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.*
- **AddProduct**  
*Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.*
- **AddWeighted**  
*Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.*
- **Mul**  
*Pixel by pixel multiply of two images.*
- **MulScale**  
*Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.*
- **Sub**  
*Pixel by pixel subtraction of two images.*
- **Div**  
*Pixel by pixel division of two images.*

- [Div\\_Round](#)  
*Pixel by pixel division of two images using result rounding modes.*
- [Abs](#)  
*Absolute value of each pixel value in an image.*
- [AbsDiff](#)  
*Pixel by pixel absolute difference between two images.*
- [Sqr](#)  
*Square each pixel in an image.*
- [Sqrt](#)  
*Pixel by pixel square root of each pixel in an image.*
- [Ln](#)  
*Pixel by pixel natural logarithm of each pixel in an image.*
- [Exp](#)  
*Exponential value of each pixel in an image.*

## 7.7 AddC

Adds a constant value to each pixel of an image.

### Functions

- `NppStatus nppiAddC_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C1IRSfs` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C3IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_AC4IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_8u_C4IRSfs` (const `Npp8u` aConstants[4], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- **NppStatus nppiAddC\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_C3IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32sc\_AC4IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- **NppStatus nppiAddC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)
 

*One 32-bit floating point channel image add constant.*
- **NppStatus nppiAddC\_32f\_C1IR** (const **Npp32f** nConstant, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
 

*One 32-bit floating point channel in place image add constant.*
- **NppStatus nppiAddC\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)
 

*Three 32-bit floating point channel image add constant.*
- **NppStatus nppiAddC\_32f\_C3IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
 

*Three 32-bit floating point channel in place image add constant.*
- **NppStatus nppiAddC\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)
 

*Four 32-bit floating point channel with unmodified alpha image add constant.*
- **NppStatus nppiAddC\_32f\_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)
 

*Four 32-bit floating point channel with unmodified alpha in place image add constant.*

*Four 32-bit floating point channel with unmodified alpha in place image add constant.*

- **NppStatus nppiAddC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel image add constant.*

- **NppStatus nppiAddC\_32f\_C4IR** (const **Npp32f** aConstants[4], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel in place image add constant.*

- **NppStatus nppiAddC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- **NppStatus nppiAddC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

- **NppStatus nppiAddC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- **NppStatus nppiAddC\_32fc\_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

- **NppStatus nppiAddC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.*

- **NppStatus nppiAddC\_32fc\_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.*

- **NppStatus nppiAddC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- **NppStatus nppiAddC\_32fc\_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

### 7.7.1 Detailed Description

Adds a constant value to each pixel of an image.

### 7.7.2 Function Documentation

#### 7.7.2.1 `NppStatus nppiAddC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.2.2 `NppStatus nppiAddC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.2.3 `NppStatus nppiAddC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.4 **NppStatus nppiAddC\_16s\_C1RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.5 **NppStatus nppiAddC\_16s\_C3IRSfs** (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.6 `NppStatus nppiAddC_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.7 `NppStatus nppiAddC_16s_C4IRSfs (const Npp16s aConstants[4], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.8 `NppStatus nppiAddC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.9 NppStatus nppiAddC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.10 NppStatus nppiAddC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.11 NppStatus nppiAddC\_16sc\_C1IRSfs (const Npp16sc nConstant, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.12 NppStatus nppiAddC\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.13 NppStatus nppiAddC\_16sc\_C3IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.14 NppStatus nppiAddC\_16sc\_C3RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.15 NppStatus nppiAddC\_16u\_AC4IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.16 NppStatus nppiAddC\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.17 NppStatus nppiAddC\_16u\_C1IRSfs (const Npp16u nConstant, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.18 NppStatus nppiAddC\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.19 NppStatus nppiAddC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.20 NppStatus nppiAddC\_16u\_C3RSfs (const Npp16u \* pSrcI, int nSrcIStep, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.21 `NppStatus nppiAddC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.22 `NppStatus nppiAddC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.23 `NppStatus nppiAddC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.24 NppStatus nppiAddC\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image add constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.25 NppStatus nppiAddC\_32f\_C1IR (const Npp32f nConstant, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image add constant.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.26 NppStatus nppiAddC\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image add constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.27 NppStatus nppiAddC\_32f\_C3IR (const Npp32f aConstants[3], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image add constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.28 NppStatus nppiAddC\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.29 NppStatus nppiAddC\_32f\_C4IR (const Npp32f aConstants[4], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image add constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.30 NppStatus nppiAddC\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[4], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.31 NppStatus nppiAddC\_32fc\_AC4IR (const Npp32fc aConstants[3], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.32 NppStatus nppiAddC\_32fc\_AC4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.33 `NppStatus nppiAddC_32fc_C1IR (const Npp32fc nConstant, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.34 `NppStatus nppiAddC_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc nConstant, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.35 `NppStatus nppiAddC_32fc_C3IR (const Npp32fc aConstants[3], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.36 NppStatus nppiAddC\_32fc\_C3R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.37 NppStatus nppiAddC\_32fc\_C4IR (const Npp32fc aConstants[4], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.38 NppStatus nppiAddC\_32fc\_C4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.39** `NppStatus nppiAddC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.40** `NppStatus nppiAddC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.41** `NppStatus nppiAddC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.42 **NppStatus nppiAddC\_32s\_C3RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.43 **NppStatus nppiAddC\_32sc\_AC4IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.44** `NppStatus nppiAddC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.45** `NppStatus nppiAddC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.46** `NppStatus nppiAddC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.

*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.47 **NppStatus nppiAddC\_32sc\_C3IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.48 **NppStatus nppiAddC\_32sc\_C3RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.49 `NppStatus nppiAddC_8u_AC4IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel..
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.50 `NppStatus nppiAddC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel..
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.51 `NppStatus nppiAddC_8u_C1IRSfs (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.52** `NppStatus nppiAddC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.53** `NppStatus nppiAddC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel..  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.54** `NppStatus nppiAddC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel..  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.55 NppStatus nppiAddC\_8u\_C4IRSfs (const Npp8u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.56 NppStatus nppiAddC\_8u\_C4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel..  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.8 MulC

Multiplies each pixel of an image by a constant value.

### Functions

- **NppStatus nppiMulC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s aConstants[4]`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16s_C4IRSfs` (const `Npp16s aConstants[4]`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C1RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc nConstant`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C1IRSfs` (const `Npp16sc nConstant`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C3RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc aConstants[3]`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C3IRSfs` (const `Npp16sc aConstants[3]`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_AC4RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc aConstants[3]`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_AC4IRSfs` (const `Npp16sc aConstants[3]`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C1RSfs` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s nConstant`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C1IRSfs` (const `Npp32s nConstant`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C3RSfs` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s aConstants[3]`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C3IRSfs` (const `Npp32s aConstants[3]`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_C1RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc nConstant`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32sc_C1IRSfs` (const `Npp32sc nConstant`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32sc_C3RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32sc_C3IRSfs` (const `Npp32sc aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32sc_AC4RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32sc_AC4IRSfs` (const `Npp32sc aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f nConstant`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel image multiply by constant.*
- `NppStatus nppiMulC_32f_C1IR` (const `Npp32f nConstant`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel in place image multiply by constant.*
- `NppStatus nppiMulC_32f_C3R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel image multiply by constant.*
- `NppStatus nppiMulC_32f_C3IR` (const `Npp32f aConstants[3]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel in place image multiply by constant.*
- `NppStatus nppiMulC_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel with unmodified alpha image multiply by constant.*

- **NppStatus nppiMulC\_32f\_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.*
- **NppStatus nppiMulC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image multiply by constant.*
- **NppStatus nppiMulC\_32f\_C4IR** (const **Npp32f** aConstants[4], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*

## 7.8.1 Detailed Description

Multiplies each pixel of an image by a constant value.

## 7.8.2 Function Documentation

### 7.8.2.1 `NppStatus nppiMulC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.2 `NppStatus nppiMulC_16s_AC4RSfs (const Npp16s * pSrcI, int nSrcIStep, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.3 NppStatus nppiMulC\_16s\_C1IRSfs (const Npp16s *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.4 NppStatus nppiMulC\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.5 NppStatus nppiMulC\_16s\_C3IRSfs (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.6 NppStatus nppiMulC\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.7 NppStatus nppiMulC\_16s\_C4IRSfs (const Npp16s aConstants[4], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.8 NppStatus nppiMulC\_16s\_C4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.9 NppStatus nppiMulC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.10 NppStatus nppiMulC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.11 `NppStatus nppiMulC_16sc_C1IRSfs (const Npp16sc nConstant, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.12 `NppStatus nppiMulC_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.13 `NppStatus nppiMulC_16sc_C3IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.14 NppStatus nppiMulC\_16sc\_C3RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.15 NppStatus nppiMulC\_16u\_AC4IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.16 NppStatus nppiMulC\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.17 NppStatus nppiMulC\_16u\_C1IRSfs (const Npp16u nConstant, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.18 NppStatus nppiMulC\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.19 NppStatus nppiMulC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.20 NppStatus nppiMulC\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.21 `NppStatus nppiMulC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.22 `NppStatus nppiMulC_16u_C4RSfs (const Npp16u * pSrcI, int nSrcIStep, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.23 `NppStatus nppiMulC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.24 NppStatus nppiMulC\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.25 NppStatus nppiMulC\_32f\_C1IR (const Npp32f nConstant, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image multiply by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.26 NppStatus nppiMulC\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.27** `NppStatus nppiMulC_32f_C3IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image multiply by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.28** `NppStatus nppiMulC_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image multiply by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.29** `NppStatus nppiMulC_32f_C4IR (const Npp32f aConstants[4], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image multiply by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.30 NppStatus nppiMulC\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[4], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.31 NppStatus nppiMulC\_32fc\_AC4IR (const Npp32fc aConstants[3], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.32 NppStatus nppiMulC\_32fc\_AC4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.33 `NppStatus nppiMulC_32fc_C1IR (const Npp32fc nConstant, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.34 `NppStatus nppiMulC_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc nConstant, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.35 `NppStatus nppiMulC_32fc_C3IR (const Npp32fc aConstants[3], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.36 `NppStatus nppiMulC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.37 `NppStatus nppiMulC_32fc_C4IR (const Npp32fc aConstants[4], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.38 `NppStatus nppiMulC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.39 NppStatus nppiMulC\_32s\_C1RSfs (const Npp32s nConstant, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.40 NppStatus nppiMulC\_32s\_C1RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.41 NppStatus nppiMulC\_32s\_C3IRSfs (const Npp32s aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.42 **NppStatus nppiMulC\_32s\_C3RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.43 **NppStatus nppiMulC\_32sc\_AC4IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.44** `NppStatus nppiMulC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.45** `NppStatus nppiMulC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.46** `NppStatus nppiMulC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.47 NppStatus nppiMulC\_32sc\_C3IRSfs (const Npp32sc aConstants[3], Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.48 NppStatus nppiMulC\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.49 `NppStatus nppiMulC_8u_AC4IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.50 `NppStatus nppiMulC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.51 `NppStatus nppiMulC_8u_C1IRSfs (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.52** `NppStatus nppiMulC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.53** `NppStatus nppiMulC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.54** `NppStatus nppiMulC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.55 `NppStatus nppiMulC_8u_C4IRSfs (const Npp8u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.56 `NppStatus nppiMulC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.9 MulCScale

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

### Functions

- **NppStatus nppiMulCScale\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*
- **NppStatus nppiMulCScale\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_C3R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u aConstants[3]`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*
- `NppStatus nppiMulCScale_16u_C3IR` (const `Npp16u aConstants[3]`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*
- `NppStatus nppiMulCScale_16u_AC4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u aConstants[3]`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.*
- `NppStatus nppiMulCScale_16u_AC4IR` (const `Npp16u aConstants[3]`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.*
- `NppStatus nppiMulCScale_16u_C4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u aConstants[4]`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*
- `NppStatus nppiMulCScale_16u_C4IR` (const `Npp16u aConstants[4]`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

## 7.9.1 Detailed Description

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

## 7.9.2 Function Documentation

### 7.9.2.1 `NppStatus nppiMulCScale_16u_AC4IR` (const `Npp16u aConstants[3]`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.2 NppStatus nppiMulCScale\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.3 NppStatus nppiMulCScale\_16u\_C1IR (const Npp16u nConstant, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.4 NppStatus nppiMulCScale\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.5 NppStatus nppiMulCScale\_16u\_C3IR (const Npp16u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.6 NppStatus nppiMulCScale\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.7 NppStatus nppiMulCScale\_16u\_C4IR (const Npp16u *aConstants*[4], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.8 NppStatus nppiMulCScale\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.9 NppStatus nppiMulCScale\_8u\_AC4IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.10 NppStatus nppiMulCScale\_8u\_AC4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.11 `NppStatus nppiMulCScale_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.12 `NppStatus nppiMulCScale_8u_C1R (const Npp8u * pSrcI, int nSrcIStep, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.13 `NppStatus nppiMulCScale_8u_C3IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.9.2.14 NppStatus nppiMulCScale\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.9.2.15 NppStatus nppiMulCScale\_8u\_C4IR (const Npp8u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.9.2.16 NppStatus nppiMulCScale\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.10 SubC

Subtracts a constant value from each pixel of an image.

### Functions

- `NppStatus nppiSubC_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_C1IRSfs` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_C3IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_AC4IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_8u_C4IRSfs` (const `Npp8u` aConstants[4], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32sc_C1RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc` `nConstant`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C1IRSfs` (const `Npp32sc` `nConstant`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C3RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc` `aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C3IRSfs` (const `Npp32sc` `aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_AC4RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc` `aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_AC4IRSfs` (const `Npp32sc` `aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f` `nConstant`, `Npp32f *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*One 32-bit floating point channel image subtract constant.*
- `NppStatus nppiSubC_32f_C1IR` (const `Npp32f` `nConstant`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*One 32-bit floating point channel in place image subtract constant.*
- `NppStatus nppiSubC_32f_C3R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f` `aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three 32-bit floating point channel image subtract constant.*
- `NppStatus nppiSubC_32f_C3IR` (const `Npp32f` `aConstants[3]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Three 32-bit floating point channel in place image subtract constant.*
- `NppStatus nppiSubC_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f` `aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four 32-bit floating point channel with unmodified alpha image subtract constant.*
- `NppStatus nppiSubC_32f_AC4IR` (const `Npp32f` `aConstants[3]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)

*Four 32-bit floating point channel with unmodified alpha in place image subtract constant.*

- `NppStatus nppiSubC_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[4]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit floating point channel image subtract constant.*

- `NppStatus nppiSubC_32f_C4IR` (const `Npp32f aConstants[4]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit floating point channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C1R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc nConstant`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C3R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_AC4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.*

- `NppStatus nppiSubC_32fc_AC4IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[4]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C4IR` (const `Npp32fc aConstants[4]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

## 7.10.1 Detailed Description

Subtracts a constant value from each pixel of an image.

## 7.10.2 Function Documentation

### 7.10.2.1 `NppStatus nppiSubC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.2 `NppStatus nppiSubC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.3 `NppStatus nppiSubC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.4 **NppStatus nppiSubC\_16s\_C1RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.5 **NppStatus nppiSubC\_16s\_C3IRSfs** (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.6 NppStatus nppiSubC\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.7 NppStatus nppiSubC\_16s\_C4IRSfs (const Npp16s aConstants[4], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.8 NppStatus nppiSubC\_16s\_C4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.9 NppStatus nppiSubC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.10 NppStatus nppiSubC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.11 NppStatus nppiSubC\_16sc\_C1IRSfs (const Npp16sc nConstant, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.12 NppStatus nppiSubC\_16sc\_C1RSfs (const Npp16sc \* pSrcI, int nSrcIStep, const Npp16sc nConstant, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.13 NppStatus nppiSubC\_16sc\_C3IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.14** `NppStatus nppiSubC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.15** `NppStatus nppiSubC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.16** `NppStatus nppiSubC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.17** `NppStatus nppiSubC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.18** `NppStatus nppiSubC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.19** `NppStatus nppiSubC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.20** `NppStatus nppiSubC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.21 NppStatus nppiSubC\_16u\_C4IRSfs (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.22 NppStatus nppiSubC\_16u\_C4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.23 NppStatus nppiSubC\_32f\_AC4IR (const Npp32f aConstants[3], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.24 NppStatus nppiSubC\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.25 NppStatus nppiSubC\_32f\_C1IR (const Npp32f nConstant, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image subtract constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.26 NppStatus nppiSubC\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.27 NppStatus nppiSubC\_32f\_C3IR (const Npp32f aConstants[3], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.28 NppStatus nppiSubC\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.29 NppStatus nppiSubC\_32f\_C4IR (const Npp32f aConstants[4], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.30 NppStatus nppiSubC\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[4], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.31 NppStatus nppiSubC\_32fc\_AC4IR (const Npp32fc aConstants[3], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.32 NppStatus nppiSubC\_32fc\_AC4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.33 NppStatus nppiSubC\_32fc\_C1IR (const Npp32fc *nConstant*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.34 NppStatus nppiSubC\_32fc\_C1R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *nConstant*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.35 NppStatus nppiSubC\_32fc\_C3IR (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.36** `NppStatus nppiSubC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.37** `NppStatus nppiSubC_32fc_C4IR (const Npp32fc aConstants[4], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.38** `NppStatus nppiSubC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.39** `NppStatus nppiSubC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.40** `NppStatus nppiSubC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.41** `NppStatus nppiSubC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.42 `NppStatus nppiSubC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.43 `NppStatus nppiSubC_32sc_AC4IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.44** `NppStatus nppiSubC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.45** `NppStatus nppiSubC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.46** `NppStatus nppiSubC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.47** `NppStatus nppiSubC_32sc_C3IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.48** `NppStatus nppiSubC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.49** `NppStatus nppiSubC_8u_AC4IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.50** `NppStatus nppiSubC_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.51** `NppStatus nppiSubC_8u_C1IRSfs (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.52 NppStatus nppiSubC\_8u\_C1RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.53 NppStatus nppiSubC\_8u\_C3RSfs (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.54 NppStatus nppiSubC\_8u\_C3RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.55 `NppStatus nppiSubC_8u_C4IRSfs (const Npp8u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.56 `NppStatus nppiSubC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.11 DivC

Divides each pixel of an image by a constant value.

### Functions

- `NppStatus nppiDivC_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_C1IRSfs` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_C3IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_AC4IRSfs` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_8u_C4IRSfs` (const `Npp8u` aConstants[4], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16u_C1IRSfs` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s aConstants[4]`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C4IRSfs` (const `Npp16s aConstants[4]`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc nConstant`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1IRSfs` (const `Npp16sc nConstant`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C3RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc aConstants[3]`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C3IRSfs` (const `Npp16sc aConstants[3]`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc aConstants[3]`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4IRSfs` (const `Npp16sc aConstants[3]`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1RSfs` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s nConstant`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1IRSfs` (const `Npp32s nConstant`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3RSfs` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32s aConstants[3]`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3IRSfs` (const `Npp32s aConstants[3]`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

- `NppStatus nppiDivC_32sc_C1RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc nConstant`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32sc_C1IRSfs` (const `Npp32sc nConstant`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32sc_C3RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32sc_C3IRSfs` (const `Npp32sc aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32sc_AC4RSfs` (const `Npp32sc *pSrc1`, int `nSrc1Step`, const `Npp32sc aConstants[3]`, `Npp32sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32sc_AC4IRSfs` (const `Npp32sc aConstants[3]`, `Npp32sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f nConstant`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel image divided by constant.*
- `NppStatus nppiDivC_32f_C1IR` (const `Npp32f nConstant`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel in place image divided by constant.*
- `NppStatus nppiDivC_32f_C3R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel image divided by constant.*
- `NppStatus nppiDivC_32f_C3IR` (const `Npp32f aConstants[3]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel in place image divided by constant.*
- `NppStatus nppiDivC_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[3]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel with unmodified alpha image divided by constant.*

- **NppStatus nppiDivC\_32f\_AC4R** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4IR** (const **Npp32f** aConstants[4], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.*
- **NppStatus nppiDivC\_32fc\_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*

### 7.11.1 Detailed Description

Divides each pixel of an image by a constant value.

### 7.11.2 Function Documentation

#### 7.11.2.1 `NppStatus nppiDivC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.2 `NppStatus nppiDivC_16s_AC4RSfs (const Npp16s * pSrcI, int nSrcIStep, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

##### Parameters:

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.3 `NppStatus nppiDivC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.4 `NppStatus nppiDivC_16s_C1RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s nConstant, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.5 `NppStatus nppiDivC_16s_C3IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.6 NppStatus nppiDivC\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.7 NppStatus nppiDivC\_16s\_C4IRSfs (const Npp16s aConstants[4], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.8 NppStatus nppiDivC\_16s\_C4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.9 NppStatus nppiDivC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.10 NppStatus nppiDivC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.11 NppStatus nppiDivC\_16sc\_C1IRSfs (const Npp16sc nConstant, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.12 NppStatus nppiDivC\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.13 NppStatus nppiDivC\_16sc\_C3IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.14** `NppStatus nppiDivC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.15** `NppStatus nppiDivC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.16** `NppStatus nppiDivC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.17** `NppStatus nppiDivC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.18** `NppStatus nppiDivC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.19** `NppStatus nppiDivC_16u_C3IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.20** `NppStatus nppiDivC_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.21 NppStatus nppiDivC\_16u\_C4IRSfs (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.22 NppStatus nppiDivC\_16u\_C4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.23 NppStatus nppiDivC\_32f\_AC4IR (const Npp32f aConstants[3], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.24** `NppStatus nppiDivC_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image divided by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.25** `NppStatus nppiDivC_32f_C1IR (const Npp32f nConstant, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image divided by constant.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.26** `NppStatus nppiDivC_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f nConstant, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image divided by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.27 NppStatus nppiDivC\_32f\_C3IR (const Npp32f aConstants[3], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image divided by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.28 NppStatus nppiDivC\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.29 NppStatus nppiDivC\_32f\_C4IR (const Npp32f aConstants[4], Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image divided by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.30 NppStatus nppiDivC\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f aConstants[4], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image divided by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.31 NppStatus nppiDivC\_32fc\_AC4IR (const Npp32fc aConstants[3], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.32 NppStatus nppiDivC\_32fc\_AC4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.33** `NppStatus nppiDivC_32fc_C1IR` (`const Npp32fc nConstant`, `Npp32fc * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.34** `NppStatus nppiDivC_32fc_C1R` (`const Npp32fc * pSrc1`, `int nSrc1Step`, `const Npp32fc nConstant`, `Npp32fc * pDst`, `int nDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.35** `NppStatus nppiDivC_32fc_C3IR` (`const Npp32fc aConstants[3]`, `Npp32fc * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.36 NppStatus nppiDivC\_32fc\_C3R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.37 NppStatus nppiDivC\_32fc\_C4IR (const Npp32fc aConstants[4], Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.38 NppStatus nppiDivC\_32fc\_C4R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.39** `NppStatus nppiDivC_32s_C1RSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.40** `NppStatus nppiDivC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.41** `NppStatus nppiDivC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.42 `NppStatus nppiDivC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.43 `NppStatus nppiDivC_32sc_AC4IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.44** `NppStatus nppiDivC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.45** `NppStatus nppiDivC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.46** `NppStatus nppiDivC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.47** `NppStatus nppiDivC_32sc_C3IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.48** `NppStatus nppiDivC_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.49 NppStatus nppiDivC\_8u\_AC4IRSfs (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.50 NppStatus nppiDivC\_8u\_AC4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.51 NppStatus nppiDivC\_8u\_C1IRSfs (const Npp8u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.52 NppStatus nppiDivC\_8u\_C1RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.53 NppStatus nppiDivC\_8u\_C3RSfs (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.54 NppStatus nppiDivC\_8u\_C3RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.55 `NppStatus nppiDivC_8u_C4IRSfs (const Npp8u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.56 `NppStatus nppiDivC_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.12 AbsDiffC

Determines absolute difference between each pixel of an image and a constant value.

### Functions

- `NppStatus nppiAbsDiffC_8u_C1R` (const `Npp8u *pSrc1`, int `nSrc1Step`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nConstant`)  
*One 8-bit unsigned char channel image absolute difference with constant.*
- `NppStatus nppiAbsDiffC_16u_C1R` (const `Npp16u *pSrc1`, int `nSrc1Step`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp16u nConstant`)  
*One 16-bit unsigned short channel image absolute difference with constant.*
- `NppStatus nppiAbsDiffC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp32f nConstant`)  
*One 32-bit floating point channel image absolute difference with constant.*

### 7.12.1 Detailed Description

Determines absolute difference between each pixel of an image and a constant value.

### 7.12.2 Function Documentation

#### 7.12.2.1 `NppStatus nppiAbsDiffC_16u_C1R` (const `Npp16u *pSrc1`, int `nSrc1Step`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp16u nConstant`)

One 16-bit unsigned short channel image absolute difference with constant.

#### Parameters:

- `pSrc1` Source-Image Pointer.
- `nSrc1Step` Source-Image Line Step.
- `nConstant` Constant.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.12.2.2 `NppStatus nppiAbsDiffC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp32f nConstant`)

One 32-bit floating point channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.12.2.3 NppStatus nppiAbsDiffC\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nConstant)**

One 8-bit unsigned char channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.13 Add

Pixel by pixel addition of two images.

### Functions

- `NppStatus nppiAdd_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_C1IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16s_AC4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16s_AC4RSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16s_C4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16s_C4IRSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_C1RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_C1IRSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_C3RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_C3IRSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_AC4RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_16sc_AC4IRSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*
- `NppStatus nppiAdd_32s_C1IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_C3IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiAdd_32sc_AC4IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{-nScaleFactor}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel image addition.*
- `NppStatus nppiAdd_32f_C1IR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel in place image addition.*
- `NppStatus nppiAdd_32f_C3R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel image addition.*
- `NppStatus nppiAdd_32f_C3IR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel in place image addition.*
- `NppStatus nppiAdd_32f_AC4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel with unmodified alpha image addition.*
- `NppStatus nppiAdd_32f_AC4IR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel with unmodified alpha in place image addition.*
- `NppStatus nppiAdd_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f *pSrc2`, int `nSrc2Step`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel image addition.*
- `NppStatus nppiAdd_32f_C4IR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel in place image addition.*
- `NppStatus nppiAdd_32fc_C1R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc *pSrc2`, int `nSrc2Step`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- `NppStatus nppiAdd_32fc_C1IR` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*
- `NppStatus nppiAdd_32fc_C3R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc *pSrc2`, int `nSrc2Step`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- `NppStatus nppiAdd_32fc_C3IR` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*

- **NppStatus nppiAdd\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.*
- **NppStatus nppiAdd\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.*
- **NppStatus nppiAdd\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- **NppStatus nppiAdd\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*

### 7.13.1 Detailed Description

Pixel by pixel addition of two images.

### 7.13.2 Function Documentation

#### 7.13.2.1 **NppStatus nppiAdd\_16s\_AC4IRSfs** (const **Npp16s** \* pSrc, int nSrcStep, **Npp16s** \* pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.13.2.2 **NppStatus nppiAdd\_16s\_AC4RSfs** (const **Npp16s** \* pSrc1, int nSrc1Step, const **Npp16s** \* pSrc2, int nSrc2Step, **Npp16s** \* pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.3 NppStatus nppiAdd\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.4 NppStatus nppiAdd\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.5 NppStatus nppiAdd\_16s\_C3IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.6 NppStatus nppiAdd\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.7 `NppStatus nppiAdd_16s_C4IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.8 `NppStatus nppiAdd_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.9 `NppStatus nppiAdd_16sc_AC4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.10 NppStatus nppiAdd\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.11 NppStatus nppiAdd\_16sc\_C1IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.12 NppStatus nppiAdd\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.13 NppStatus nppiAdd\_16sc\_C3RSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.14 NppStatus nppiAdd\_16sc\_C3RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.15 **NppStatus nppiAdd\_16u\_AC4IRSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.16 **NppStatus nppiAdd\_16u\_AC4RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.17 NppStatus nppiAdd\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.18 NppStatus nppiAdd\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.19** `NppStatus nppiAdd_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.20** `NppStatus nppiAdd_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.21** `NppStatus nppiAdd_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.22** `NppStatus nppiAdd_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.23** `NppStatus nppiAdd_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.24 NppStatus nppiAdd\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.25 NppStatus nppiAdd\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.26 NppStatus nppiAdd\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.27** `NppStatus nppiAdd_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.28** `NppStatus nppiAdd_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.29 NppStatus nppiAdd\_32f\_C4IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.30 NppStatus nppiAdd\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.31 NppStatus nppiAdd\_32fc\_AC4IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.32** `NppStatus nppiAdd_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.33** `NppStatus nppiAdd_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.34** `NppStatus nppiAdd_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.235** `NppStatus nppiAdd_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.236** `NppStatus nppiAdd_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.37** `NppStatus nppiAdd_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.38** `NppStatus nppiAdd_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.39** `NppStatus nppiAdd_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.40 NppStatus nppiAdd\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image add. Add the pixel values of corresponding pixels in the ROI and write them to the output image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.41 NppStatus nppiAdd\_32s\_C1RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.42** `NppStatus nppiAdd_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.43** `NppStatus nppiAdd_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.44** `NppStatus nppiAdd_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.45 NppStatus nppiAdd\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.46 NppStatus nppiAdd\_32sc\_C1IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.47** `NppStatus nppiAdd_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.48** `NppStatus nppiAdd_32sc_C3RSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.49** `NppStatus nppiAdd_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.50 `NppStatus nppiAdd_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.51 `NppStatus nppiAdd_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.52** `NppStatus nppiAdd_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.53** `NppStatus nppiAdd_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.54** `NppStatus nppiAdd_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.55** `NppStatus nppiAdd_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.56** `NppStatus nppiAdd_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.57** `NppStatus nppiAdd_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.14 AddSquare

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

### Functions

- **NppStatus nppiAddSquare\_8u32f\_C1IMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_8u32f\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image squared then added to in place floating point destination image.*
- **NppStatus nppiAddSquare\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_16u32f\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image squared then added to in place floating point destination image.*
- **NppStatus nppiAddSquare\_32f\_C1IMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image squared then added to in place floating point destination image.*

### 7.14.1 Detailed Description

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

### 7.14.2 Function Documentation

#### 7.14.2.1 **NppStatus nppiAddSquare\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.14.2.2 **NppStatus nppiAddSquare\_16u32f\_C1IR** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.14.2.3 **NppStatus nppiAddSquare\_32f\_C1IMR** (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.14.2.4 NppStatus nppiAddSquare\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.14.2.5 NppStatus nppiAddSquare\_8u32f\_C1IMR (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.14.2.6 NppStatus nppiAddSquare\_8u32f\_C1IR (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.15 AddProduct

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

### Functions

- `NppStatus nppiAddProduct_8u32f_C1IMR` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- `NppStatus nppiAddProduct_8u32f_C1IR` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image product added to in place floating point destination image.*
- `NppStatus nppiAddProduct_16u32f_C1IMR` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- `NppStatus nppiAddProduct_16u32f_C1IR` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image product added to in place floating point destination image.*
- `NppStatus nppiAddProduct_32f_C1IMR` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- `NppStatus nppiAddProduct_32f_C1IR` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image product added to in place floating point destination image.*

### 7.15.1 Detailed Description

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

### 7.15.2 Function Documentation

- 7.15.2.1 `NppStatus nppiAddProduct_16u32f_C1IMR`** (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.2 NppStatus nppiAddProduct\_16u32f\_C11R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.3 NppStatus nppiAddProduct\_32f\_C11MR (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.4 NppStatus nppiAddProduct\_32f\_C1IR (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit floating point channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.5 NppStatus nppiAddProduct\_8u32f\_C1IMR (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.15.2.6 NppStatus nppiAddProduct\_8u32f\_C1IR (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image product added to in place floating point destination image.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.16 AddWeighted

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

### Functions

- **NppStatus nppiAddWeighted\_8u32f\_C1IMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddWeighted\_8u32f\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.*
- **NppStatus nppiAddWeighted\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddWeighted\_16u32f\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.*
- **NppStatus nppiAddWeighted\_32f\_C1IMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddWeighted\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 32-bit floating point channel alpha weighted image added to in place floating point destination image.*

### 7.16.1 Detailed Description

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

### 7.16.2 Function Documentation

#### 7.16.2.1 **NppStatus nppiAddWeighted\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.16.2.2 NppStatus nppiAddWeighted\_16u32f\_C1IR (const Npp16u \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.16.2.3 NppStatus nppiAddWeighted\_32f\_C1IMR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.4 NppStatus nppiAddWeighted\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)**

One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.5 NppStatus nppiAddWeighted\_8u32f\_C1IMR (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)**

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.6 NppStatus nppiAddWeighted\_8u32f\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, Npp32f *nAlpha*)**

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.17 Mul

Pixel by pixel multiply of two images.

### Functions

- **NppStatus nppiMul\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16u_C1RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C3RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_AC4RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C4RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C1RSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C3RSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16s_AC4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16s_AC4RSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16s_C4RSfs` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16s_C4RSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_C1RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_C1RSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_C3RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_C3RSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_AC4RSfs` (const `Npp16sc *pSrc1`, int `nSrc1Step`, const `Npp16sc *pSrc2`, int `nSrc2Step`, `Npp16sc *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16sc_AC4RSfs` (const `Npp16sc *pSrc`, int `nSrcStep`, `Npp16sc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*
- `NppStatus nppiMul_32s_C1IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32sc_C3IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32sc_AC4IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C1IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C3IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image multiplication.*
- `NppStatus nppiMul_32f_AC4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image multiplication.*
- `NppStatus nppiMul_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- `NppStatus nppiMul_32fc_C1IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- `NppStatus nppiMul_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*

- **NppStatus nppiMul\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.*
- **NppStatus nppiMul\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- **NppStatus nppiMul\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*

### 7.17.1 Detailed Description

Pixel by pixel multiply of two images.

### 7.17.2 Function Documentation

#### 7.17.2.1 **NppStatus nppiMul\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.2 NppStatus nppiMul\_16s\_AC4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.3 NppStatus nppiMul\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.4 NppStatus nppiMul\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.17.2.5 NppStatus nppiMul\_16s\_C3IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.17.2.6 NppStatus nppiMul\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.7 NppStatus nppiMul\_16s\_C4IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.8 NppStatus nppiMul\_16s\_C4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.9 NppStatus nppiMul\_16sc\_AC4IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.10 **NppStatus nppiMul\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.11 **NppStatus nppiMul\_16sc\_C1IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.12 NppStatus nppiMul\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.13 NppStatus nppiMul\_16sc\_C3RSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.14 NppStatus nppiMul\_16sc\_C3RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.15 **NppStatus nppiMul\_16u\_AC4IRSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.16 **NppStatus nppiMul\_16u\_AC4RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.17 NppStatus nppiMul\_16u\_C1IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.18 NppStatus nppiMul\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.19** `NppStatus nppiMul_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.20** `NppStatus nppiMul_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.21** `NppStatus nppiMul_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.22** `NppStatus nppiMul_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.23** `NppStatus nppiMul_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.24 NppStatus nppiMul\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.25 NppStatus nppiMul\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.26 NppStatus nppiMul\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.27** `NppStatus nppiMul_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.28** `NppStatus nppiMul_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.29 NppStatus nppiMul\_32f\_C4IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.30 NppStatus nppiMul\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.31 NppStatus nppiMul\_32fc\_AC4IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.32** `NppStatus nppiMul_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.33** `NppStatus nppiMul_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.34** `NppStatus nppiMul_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.35 NppStatus nppiMul\_32fc\_C3IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.36 NppStatus nppiMul\_32fc\_C3R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pSrc2, int nSrc2Step, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.37** `NppStatus nppiMul_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.38** `NppStatus nppiMul_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.39** `NppStatus nppiMul_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.40 NppStatus nppiMul\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

1 channel 32-bit image multiplication. Multiply corresponding pixels in ROI.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.41 NppStatus nppiMul\_32s\_C1RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.42** `NppStatus nppiMul_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.43** `NppStatus nppiMul_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.44** `NppStatus nppiMul_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.45 NppStatus nppiMul\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.46 NppStatus nppiMul\_32sc\_C1IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.47** `NppStatus nppiMul_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.48** `NppStatus nppiMul_32sc_C3IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.17.2.49** `NppStatus nppiMul_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.50 **NppStatus nppiMul\_8u\_AC4IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.51 **NppStatus nppiMul\_8u\_AC4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.52** `NppStatus nppiMul_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.53** `NppStatus nppiMul_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.54** `NppStatus nppiMul_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.55** `NppStatus nppiMul_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.56** `NppStatus nppiMul_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.57** `NppStatus nppiMul_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.18 MulScale

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

### Functions

- `NppStatus nppiMulScale_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*
- `NppStatus nppiMulScale_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_C3R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_C3IR` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_AC4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_AC4IR` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_C4R` (const `Npp16u *pSrc1`, int `nSrc1Step`, const `Npp16u *pSrc2`, int `nSrc2Step`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- `NppStatus nppiMulScale_16u_C4IR` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

## 7.18.1 Detailed Description

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

## 7.18.2 Function Documentation

### 7.18.2.1 `NppStatus nppiMulScale_16u_AC4IR` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.2 NppStatus nppiMulScale\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.3 NppStatus nppiMulScale\_16u\_C11R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.18.2.4 `NppStatus nppiMulScale_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.5 `NppStatus nppiMulScale_16u_C3IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.6 `NppStatus nppiMulScale_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.7 NppStatus nppiMulScale\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.8 NppStatus nppiMulScale\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.9** `NppStatus nppiMulScale_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.18.2.10** `NppStatus nppiMulScale_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.18.2.11** `NppStatus nppiMulScale_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.12** `NppStatus nppiMulScale_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.13** `NppStatus nppiMulScale_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.14** `NppStatus nppiMulScale_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.18.2.15** `NppStatus nppiMulScale_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.18.2.16** `NppStatus nppiMulScale_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.19 Sub

Pixel by pixel subtraction of two images.

### Functions

- `NppStatus nppiSub_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C1IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)
 

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*
- **NppStatus nppiSub\_32s\_C1IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C3IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32s\_C4IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- [NppStatus nppiSub\\_32sc\\_C3IRSfs](#) (const [Npp32sc](#) \*pSrc, int nSrcStep, [Npp32sc](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)
 

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- [NppStatus nppiSub\\_32sc\\_AC4RSfs](#) (const [Npp32sc](#) \*pSrc1, int nSrc1Step, const [Npp32sc](#) \*pSrc2, int nSrc2Step, [Npp32sc](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)
 

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- [NppStatus nppiSub\\_32sc\\_AC4IRSfs](#) (const [Npp32sc](#) \*pSrc, int nSrcStep, [Npp32sc](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)
 

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- [NppStatus nppiSub\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)
 

*One 32-bit floating point channel image subtraction.*
- [NppStatus nppiSub\\_32f\\_C1IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
 

*One 32-bit floating point channel in place image subtraction.*
- [NppStatus nppiSub\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)
 

*Three 32-bit floating point channel image subtraction.*
- [NppStatus nppiSub\\_32f\\_C3IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
 

*One 32-bit floating point channel in place image subtraction.*
- [NppStatus nppiSub\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)
 

*Four 32-bit floating point channel with unmodified alpha image subtraction.*
- [NppStatus nppiSub\\_32f\\_AC4IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
 

*Four 32-bit floating point channel with unmodified alpha in place image subtraction.*
- [NppStatus nppiSub\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)
 

*Four 32-bit floating point channel image subtraction.*
- [NppStatus nppiSub\\_32f\\_C4IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)
 

*Four 32-bit floating point channel in place image subtraction.*
- [NppStatus nppiSub\\_32fc\\_C1R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [Npp32fc](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)
 

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*

- **NppStatus nppiSub\_32fc\_C1IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*
- **NppStatus nppiSub\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*
- **NppStatus nppiSub\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*
- **NppStatus nppiSub\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.*
- **NppStatus nppiSub\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.*
- **NppStatus nppiSub\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*
- **NppStatus nppiSub\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*

### 7.19.1 Detailed Description

Pixel by pixel subtraction of two images.

### 7.19.2 Function Documentation

#### 7.19.2.1 **NppStatus nppiSub\_16s\_AC4IRSfs** (const **Npp16s** \* pSrc, int nSrcStep, **Npp16s** \* pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.2** `NppStatus nppiSub_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.3** `NppStatus nppiSub_16s_C1IRSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.4 NppStatus nppiSub\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.5 NppStatus nppiSub\_16s\_C3RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.6 NppStatus nppiSub\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.7 **NppStatus nppiSub\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.8 **NppStatus nppiSub\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.9 NppStatus nppiSub\_16sc\_AC4IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.10 NppStatus nppiSub\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.11 NppStatus nppiSub\_16sc\_C1IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.12** `NppStatus nppiSub_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.13** `NppStatus nppiSub_16sc_C3IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.14** `NppStatus nppiSub_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.15** `NppStatus nppiSub_16u_AC4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.16** `NppStatus nppiSub_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.17 **NppStatus nppiSub\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.18 **NppStatus nppiSub\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.19 NppStatus nppiSub\_16u\_C3IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.20 NppStatus nppiSub\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.21** `NppStatus nppiSub_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.22** `NppStatus nppiSub_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.23** `NppStatus nppiSub_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.24 **NppStatus nppiSub\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel with unmodified alpha image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.25 **NppStatus nppiSub\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.26** `NppStatus nppiSub_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.27** `NppStatus nppiSub_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.28** `NppStatus nppiSub_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.29** `NppStatus nppiSub_32f_C4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.30** `NppStatus nppiSub_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.31** `NppStatus nppiSub_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.32** `NppStatus nppiSub_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.33** `NppStatus nppiSub_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.34** `NppStatus nppiSub_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.35** `NppStatus nppiSub_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.36** `NppStatus nppiSub_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.37** `NppStatus nppiSub_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.38** `NppStatus nppiSub_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.39 NppStatus nppiSub\_32s\_C1IRSfs (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.40 NppStatus nppiSub\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image subtraction. Subtract pSrc1's pixels from corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.41** `NppStatus nppiSub_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.42** `NppStatus nppiSub_32s_C3IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.43** `NppStatus nppiSub_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.44 **NppStatus nppiSub\_32s\_C4IRSfs** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.45 **NppStatus nppiSub\_32s\_C4RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.46 NppStatus nppiSub\_32sc\_AC4IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.47 NppStatus nppiSub\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.48** `NppStatus nppiSub_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.49** `NppStatus nppiSub_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.50** `NppStatus nppiSub_32sc_C3IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.51 **NppStatus nppiSub\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.52 **NppStatus nppiSub\_8u\_AC4IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.53** `NppStatus nppiSub_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.54** `NppStatus nppiSub_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.55** `NppStatus nppiSub_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.56 NppStatus nppiSub\_8u\_C3IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.19.2.57 NppStatus nppiSub\_8u\_C3RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 8-bit unsigned char channel image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.19.2.58** `NppStatus nppiSub_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.59** `NppStatus nppiSub_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.20 Div

Pixel by pixel division of two images.

### Functions

- `NppStatus nppiDiv_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_C1IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiDiv_16u_C1IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- `NppStatus nppiDiv_32s_C1IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_C3IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiDiv_32sc_AC4IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- [NppStatus nppiDiv\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point channel image division.*
- [NppStatus nppiDiv\\_32f\\_C1IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point channel in place image division.*
- [NppStatus nppiDiv\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three 32-bit floating point channel image division.*
- [NppStatus nppiDiv\\_32f\\_C3IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point channel in place image division.*
- [NppStatus nppiDiv\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image division.*
- [NppStatus nppiDiv\\_32f\\_AC4IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image division.*
- [NppStatus nppiDiv\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four 32-bit floating point channel image division.*
- [NppStatus nppiDiv\\_32f\\_C4IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four 32-bit floating point channel in place image division.*
- [NppStatus nppiDiv\\_32fc\\_C1R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [Npp32fc](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- [NppStatus nppiDiv\\_32fc\\_C1IR](#) (const [Npp32fc](#) \*pSrc, int nSrcStep, [Npp32fc](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- [NppStatus nppiDiv\\_32fc\\_C3R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [Npp32fc](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- [NppStatus nppiDiv\\_32fc\\_C3IR](#) (const [Npp32fc](#) \*pSrc, int nSrcStep, [Npp32fc](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- [NppStatus nppiDiv\\_32fc\\_AC4R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [Npp32fc](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

- **NppStatus nppiDiv\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

- **NppStatus nppiDiv\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

- **NppStatus nppiDiv\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

## 7.20.1 Detailed Description

Pixel by pixel division of two images.

## 7.20.2 Function Documentation

### 7.20.2.1 NppStatus nppiDiv\_16s\_AC4IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.2 NppStatus nppiDiv\_16s\_AC4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.3 NppStatus nppiDiv\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.4 NppStatus nppiDiv\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.5 NppStatus nppiDiv\_16s\_C3IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.6 NppStatus nppiDiv\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.7 NppStatus nppiDiv\_16s\_C4IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.8 `NppStatus nppiDiv_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.9 `NppStatus nppiDiv_16sc_AC4IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.10 NppStatus nppiDiv\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.11 NppStatus nppiDiv\_16sc\_C1IRSfs (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.12 NppStatus nppiDiv\_16sc\_C1RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc \* pSrc2, int nSrc2Step, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.13 `NppStatus nppiDiv_16sc_C3IRSfs (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.14 `NppStatus nppiDiv_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.15 NppStatus nppiDiv\_16u\_AC4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.16 NppStatus nppiDiv\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.17** `NppStatus nppiDiv_16u_C1IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.18** `NppStatus nppiDiv_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.19** `NppStatus nppiDiv_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.20** `NppStatus nppiDiv_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.21** `NppStatus nppiDiv_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.22** `NppStatus nppiDiv_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.23** `NppStatus nppiDiv_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.24** `NppStatus nppiDiv_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.25** `NppStatus nppiDiv_32f_C1IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.26** `NppStatus nppiDiv_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.27 NppStatus nppiDiv\_32f\_C3IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.28 NppStatus nppiDiv\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.29 NppStatus nppiDiv\_32f\_C4IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.30** `NppStatus nppiDiv_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.31** `NppStatus nppiDiv_32fc_AC4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.32** `NppStatus nppiDiv_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.33** `NppStatus nppiDiv_32fc_C1IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.34** `NppStatus nppiDiv_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.35** `NppStatus nppiDiv_32fc_C3IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.36** `NppStatus nppiDiv_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.37** `NppStatus nppiDiv_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.38** `NppStatus nppiDiv_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.39** `NppStatus nppiDiv_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.40** `NppStatus nppiDiv_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image division. Divide pixels in pSrc2 by pSrc1's pixels.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.41 NppStatus nppiDiv\_32s\_C1RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.42 NppStatus nppiDiv\_32s\_C3IRSfs (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.43** `NppStatus nppiDiv_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.44** `NppStatus nppiDiv_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.45** `NppStatus nppiDiv_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.46 `NppStatus nppiDiv_32sc_C1RSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.47 `NppStatus nppiDiv_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.48 NppStatus nppiDiv\_32sc\_C3IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.49 NppStatus nppiDiv\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.50** `NppStatus nppiDiv_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.51** `NppStatus nppiDiv_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.52** `NppStatus nppiDiv_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.53 `NppStatus nppiDiv_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.54 `NppStatus nppiDiv_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.55** `NppStatus nppiDiv_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.56** `NppStatus nppiDiv_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.57** `NppStatus nppiDiv_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.21 Div\_Round

Pixel by pixel division of two images using result rounding modes.

### Functions

- **NppStatus nppiDiv\_Round\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

### 7.21.1 Detailed Description

Pixel by pixel division of two images using result rounding modes.

### 7.21.2 Function Documentation

#### 7.21.2.1 **NppStatus nppiDiv\_Round\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.2 NppStatus nppiDiv\_Round\_16s\_AC4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.3 NppStatus nppiDiv\_Round\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.4 NppStatus nppiDiv\_Round\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.5 NppStatus nppiDiv\_Round\_16s\_C3IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.6 NppStatus nppiDiv\_Round\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.7 NppStatus nppiDiv\_Round\_16s\_C4IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.8 NppStatus nppiDiv\_Round\_16s\_C4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.9 NppStatus nppiDiv\_Round\_16u\_AC4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.10** `NppStatus nppiDiv_Round_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.11** `NppStatus nppiDiv_Round_16u_C1IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.12** `NppStatus nppiDiv_Round_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.13** `NppStatus nppiDiv_Round_16u_C3IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.14 NppStatus nppiDiv\_Round\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.15 NppStatus nppiDiv\_Round\_16u\_C4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.16** `NppStatus nppiDiv_Round_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.17** `NppStatus nppiDiv_Round_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.18** `NppStatus nppiDiv_Round_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.19** `NppStatus nppiDiv_Round_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.20** `NppStatus nppiDiv_Round_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.21** `NppStatus nppiDiv_Round_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 8-bit unsigned char channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.22 NppStatus nppiDiv\_Round\_8u\_C3RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.23 NppStatus nppiDiv\_Round\_8u\_C4IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)**

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.24** `NppStatus nppiDiv_Round_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.22 Abs

Absolute value of each pixel value in an image.

### Functions

- `NppStatus nppiAbs_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image absolute value with unmodified alpha.*
- `NppStatus nppiAbs_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image absolute value with unmodified alpha.*
- `NppStatus nppiAbs_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image absolute value.*
- `NppStatus nppiAbs_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image absolute value.*
- `NppStatus nppiAbs_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image absolute value.*
- `NppStatus nppiAbs_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel in place image absolute value.*
- `NppStatus nppiAbs_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel image absolute value with unmodified alpha.*

- **NppStatus nppiAbs\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value with unmodified alpha.*
- **NppStatus nppiAbs\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image absolute value.*
- **NppStatus nppiAbs\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value.*

### 7.22.1 Detailed Description

Absolute value of each pixel value in an image.

### 7.22.2 Function Documentation

#### 7.22.2.1 NppStatus nppiAbs\_16s\_AC4IR (Npp16s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel in place image absolute value with unmodified alpha.

##### Parameters:

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.22.2.2 NppStatus nppiAbs\_16s\_AC4R (const Npp16s \*pSrc, int nSrcStep, Npp16s \*pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel image absolute value with unmodified alpha.

##### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.3 NppStatus nppiAbs\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.4 NppStatus nppiAbs\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.5 NppStatus nppiAbs\_16s\_C3IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.6 NppStatus nppiAbs\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.7 NppStatus nppiAbs\_16s\_C4IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.8 NppStatus nppiAbs\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.9 NppStatus nppiAbs\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image absolute value with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.10 NppStatus nppiAbs\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image absolute value with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.11 NppStatus nppiAbs\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.12 NppStatus nppiAbs\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.13 NppStatus nppiAbs\_32f\_C3IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.14 NppStatus nppiAbs\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.15 NppStatus nppiAbs\_32f\_C4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.16 NppStatus nppiAbs\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.23 AbsDiff

Pixel by pixel absolute difference between two images.

### Functions

- `NppStatus nppiAbsDiff_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel absolute difference of image1 minus image2.*
- `NppStatus nppiAbsDiff_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channels absolute difference of image1 minus image2.*
- `NppStatus nppiAbsDiff_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channels absolute difference of image1 minus image2.*
- `NppStatus nppiAbsDiff_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel absolute difference of image1 minus image2.*
- `NppStatus nppiAbsDiff_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel absolute difference of image1 minus image2.*

### 7.23.1 Detailed Description

Pixel by pixel absolute difference between two images.

### 7.23.2 Function Documentation

#### 7.23.2.1 `NppStatus nppiAbsDiff_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

One 16-bit unsigned short channel absolute difference of image1 minus image2.

#### Parameters:

- `pSrc1` Source-Image Pointer.
- `nSrc1Step` Source-Image Line Step.
- `pSrc2` Source-Image Pointer.
- `nSrc2Step` Source-Image Line Step.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.2** `NppStatus nppiAbsDiff_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.3** `NppStatus nppiAbsDiff_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.4** `NppStatus nppiAbsDiff_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channels absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.23.2.5 NppStatus nppiAbsDiff\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channels absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.24 Sqr

Square each pixel in an image.

### Functions

- **NppStatus nppiSqr\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

*One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSqr_16s_AC4RSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_AC4IRSfs` (`Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_C4RSfs` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_C4IRSfs` (`Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, int `nScaleFactor`)  
*Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C1IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit floating point channel in place image squared.*
- `NppStatus nppiSqr_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C3IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit floating point channel in place image squared.*
- `NppStatus nppiSqr_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel image squared with unmodified alpha.*
- `NppStatus nppiSqr_32f_AC4IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel in place image squared with unmodified alpha.*
- `NppStatus nppiSqr_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C4IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 32-bit floating point channel in place image squared.*

### 7.24.1 Detailed Description

Square each pixel in an image.

## 7.24.2 Function Documentation

### 7.24.2.1 `NppStatus nppiSqr_16s_AC4IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.2 `NppStatus nppiSqr_16s_AC4RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.3 `NppStatus nppiSqr_16s_C1IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.24.2.4 `NppStatus nppiSqr_16s_C1RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel image squared, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.5 `NppStatus nppiSqr_16s_C3IRSfs (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel in place image squared, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.6 `NppStatus nppiSqr_16s_C3RSfs (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image squared, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.7 NppStatus nppiSqr\_16s\_C4IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.24.2.8 NppStatus nppiSqr\_16s\_C4RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.24.2.9 NppStatus nppiSqr\_16u\_AC4IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.24.2.10** `NppStatus nppiSqr_16u_AC4RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.11** `NppStatus nppiSqr_16u_C1RSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.12** `NppStatus nppiSqr_16u_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.13** `NppStatus nppiSqr_16u_C3IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.14** `NppStatus nppiSqr_16u_C3RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.15** `NppStatus nppiSqr_16u_C4IRSfs (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.16 NppStatus nppiSqr\_16u\_C4RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image squared, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.17 NppStatus nppiSqr\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image squared with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.18 NppStatus nppiSqr\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image squared with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.19 NppStatus nppiSqr\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.20 NppStatus nppiSqr\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.21 NppStatus nppiSqr\_32f\_C3IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.22 NppStatus nppiSqr\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image squared.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.23 NppStatus nppiSqr\_32f\_C4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image squared.

**Parameters:**

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.24 NppStatus nppiSqr\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image squared.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.25 NppStatus nppiSqr\_8u\_AC4IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.26 NppStatus nppiSqr\_8u\_AC4RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.27 NppStatus nppiSqr\_8u\_C1IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.28** `NppStatus nppiSqr_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.29** `NppStatus nppiSqr_8u_C3IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.30** `NppStatus nppiSqr_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.31 NppStatus nppiSqr\_8u\_C4IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.32 NppStatus nppiSqr\_8u\_C4RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.25 Sqrt

Pixel by pixel square root of each pixel in an image.

### Functions

- **NppStatus nppiSqrt\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_AC4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_AC4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image square root.*
- **NppStatus nppiSqrt\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image square root.*

- [NppStatus nppiSqrt\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Three 32-bit floating point channel image square root.*

- [NppStatus nppiSqrt\\_32f\\_C3IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Three 32-bit floating point channel in place image square root.*

- [NppStatus nppiSqrt\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel image square root with unmodified alpha.*

- [NppStatus nppiSqrt\\_32f\\_AC4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel in place image square root with unmodified alpha.*

- [NppStatus nppiSqrt\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel image square root.*

- [NppStatus nppiSqrt\\_32f\\_C4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel in place image square root.*

## 7.25.1 Detailed Description

Pixel by pixel square root of each pixel in an image.

## 7.25.2 Function Documentation

### 7.25.2.1 [NppStatus nppiSqrt\\_16s\\_AC4IRSfs](#) ([Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.25.2.2 NppStatus nppiSqrt\_16s\_AC4RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.3 NppStatus nppiSqrt\_16s\_C1IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.4 NppStatus nppiSqrt\_16s\_C1RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.5 NppStatus nppiSqrt\_16s\_C3IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.6 NppStatus nppiSqrt\_16s\_C3RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.7 NppStatus nppiSqrt\_16u\_AC4IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.8 NppStatus nppiSqrt\_16u\_AC4RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.9 NppStatus nppiSqrt\_16u\_C1IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.10 NppStatus nppiSqrt\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.11 NppStatus nppiSqrt\_16u\_C3IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.12 NppStatus nppiSqrt\_16u\_C3RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.25.2.13 NppStatus nppiSqrt\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image square root with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.14 NppStatus nppiSqrt\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image square root with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.15 NppStatus nppiSqrt\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.16** `NppStatus nppiSqrt_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image square root.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.17** `NppStatus nppiSqrt_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image square root.

**Parameters:**

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.18** `NppStatus nppiSqrt_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image square root.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.19 NppStatus nppiSqrt\_32f\_C4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.20 NppStatus nppiSqrt\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.21 NppStatus nppiSqrt\_8u\_AC4IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.22 NppStatus nppiSqrt\_8u\_AC4RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.23 NppStatus nppiSqrt\_8u\_C1IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.24 NppStatus nppiSqrt\_8u\_C1RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.25** `NppStatus nppiSqrt_8u_C3IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel in place image square root, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.26** `NppStatus nppiSqrt_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image square root, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.26 Ln

Pixel by pixel natural logarithm of each pixel in an image.

### Functions

- **NppStatus nppiLn\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)
 

*One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiLn\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image natural logarithm.*
- **NppStatus nppiLn\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image natural logarithm.*
- **NppStatus nppiLn\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image natural logarithm.*
- **NppStatus nppiLn\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image natural logarithm.*

## 7.26.1 Detailed Description

Pixel by pixel natural logarithm of each pixel in an image.

## 7.26.2 Function Documentation

### 7.26.2.1 **NppStatus nppiLn\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrcDst** In-Place Image Pointer.
- nSrcDstStep** In-Place-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).
- nScaleFactor** Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.2 **NppStatus nppiLn\_16s\_C1RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.3 **NppStatus nppiLn\_16s\_C3IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.4 **NppStatus nppiLn\_16s\_C3RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.5 NppStatus nppiLn\_16u\_C1IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.6 NppStatus nppiLn\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.7 NppStatus nppiLn\_16u\_C3IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.8 `NppStatus nppiLn_16u_C3RSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.9 `NppStatus nppiLn_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image natural logarithm.

#### Parameters:

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.10 `NppStatus nppiLn_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image natural logarithm.

#### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.11 NppStatus nppiLn\_32f\_C3IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel in place image natural logarithm.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.12 NppStatus nppiLn\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image natural logarithm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.13 NppStatus nppiLn\_8u\_C1IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.14** `NppStatus nppiLn_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.26.2.15** `NppStatus nppiLn_8u_C3IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.26.2.16** `NppStatus nppiLn_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.27 Exp

Exponential value of each pixel in an image.

### Functions

- **NppStatus nppiExp\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiExp\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image exponential.*
- **NppStatus nppiExp\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image exponential.*

## 7.27.1 Detailed Description

Exponential value of each pixel in an image.

## 7.27.2 Function Documentation

### 7.27.2.1 **NppStatus nppiExp\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.2 NppStatus nppiExp\_16s\_C1RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.3 NppStatus nppiExp\_16s\_C3IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.4 NppStatus nppiExp\_16s\_C3RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.5 NppStatus nppiExp\_16u\_C1IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.27.2.6 NppStatus nppiExp\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.27.2.7 NppStatus nppiExp\_16u\_C3IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.8 NppStatus nppiExp\_16u\_C3RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.9 NppStatus nppiExp\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image exponential.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.10 NppStatus nppiExp\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image exponential.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.11 `NppStatus nppiExp_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image exponential.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.12 `NppStatus nppiExp_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image exponential.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.13 `NppStatus nppiExp_8u_C1IRSfs (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.14 NppStatus nppiExp\_8u\_C1RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.15 NppStatus nppiExp\_8u\_C3IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.16 NppStatus nppiExp\_8u\_C3RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.28 Logical Operations

### Modules

- [AndC](#)  
*Pixel by pixel logical and of an image with a constant.*
- [OrC](#)  
*Pixel by pixel logical or of an image with a constant.*
- [XorC](#)  
*Pixel by pixel logical exclusive or of an image with a constant.*
- [RShiftC](#)  
*Pixel by pixel right shift of an image by a constant value.*
- [LShiftC](#)  
*Pixel by pixel left shift of an image by a constant value.*
- [And](#)  
*Pixel by pixel logical and of images.*
- [Or](#)  
*Pixel by pixel logical or of images.*
- [Xor](#)  
*Pixel by pixel logical exclusive or of images.*
- [Not](#)  
*Pixel by pixel logical not of image.*

## 7.29 AndC

Pixel by pixel logical and of an image with a constant.

### Functions

- `NppStatus nppiAndC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image logical and with constant.*
- `NppStatus nppiAndC_8u_C1IR` (const `Npp8u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image logical and with constant.*
- `NppStatus nppiAndC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image logical and with constant.*
- `NppStatus nppiAndC_8u_C3IR` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and with constant.*
- `NppStatus nppiAndC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_8u_AC4IR` (const `Npp8u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` aConstants[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant.*
- `NppStatus nppiAndC_8u_C4IR` (const `Npp8u` aConstants[4], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant.*
- `NppStatus nppiAndC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` nConstant, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image logical and with constant.*
- `NppStatus nppiAndC_16u_C1IR` (const `Npp16u` nConstant, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image logical and with constant.*

- `NppStatus nppiAndC_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and with constant.*
- `NppStatus nppiAndC_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C3IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_AC4IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C4IR` (const `Npp32s` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with constant.*

### 7.29.1 Detailed Description

Pixel by pixel logical and of an image with a constant.

### 7.29.2 Function Documentation

#### 7.29.2.1 `NppStatus nppiAndC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.

##### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.29.2.2 `NppStatus nppiAndC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.29.2.3 `NppStatus nppiAndC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and with constant.

##### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.4 NppStatus nppiAndC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.5 NppStatus nppiAndC\_16u\_C3IR (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.6 NppStatus nppiAndC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.7 NppStatus nppiAndC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.8 NppStatus nppiAndC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.9 `NppStatus nppiAndC_32s_AC4IR` (`const Npp32s aConstants[3]`, `Npp32s * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.10 `NppStatus nppiAndC_32s_AC4R` (`const Npp32s * pSrc1`, `int nSrc1Step`, `const Npp32s aConstants[3]`, `Npp32s * pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Four 32-bit signed integer channel image logical and with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.11 `NppStatus nppiAndC_32s_C1IR` (`const Npp32s nConstant`, `Npp32s * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit signed integer channel in place image logical and with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.12 NppStatus nppiAndC\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.13 NppStatus nppiAndC\_32s\_C3IR (const Npp32s aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.14 NppStatus nppiAndC\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.15 NppStatus nppiAndC\_32s\_C4IR (const Npp32s aConstants[4], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.16 NppStatus nppiAndC\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s aConstants[4], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.17 NppStatus nppiAndC\_8u\_AC4IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.18** `NppStatus nppiAndC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.19** `NppStatus nppiAndC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical and with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.20** `NppStatus nppiAndC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.21 NppStatus nppiAndC\_8u\_C3IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.22 NppStatus nppiAndC\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.23 NppStatus nppiAndC\_8u\_C4IR (const Npp8u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.29.2.24 NppStatus nppiAndC\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.30 OrC

Pixel by pixel logical or of an image with a constant.

### Functions

- **NppStatus nppiOrC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical or with constant.*

- `NppStatus nppiOrC_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical or with constant.*
- `NppStatus nppiOrC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant.*
- `NppStatus nppiOrC_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C3IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_AC4IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C4IR` (const `Npp32s` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant.*

### 7.30.1 Detailed Description

Pixel by pixel logical or of an image with a constant.

### 7.30.2 Function Documentation

#### 7.30.2.1 `NppStatus nppiOrC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.30.2.2 `NppStatus nppiOrC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.30.2.3 `NppStatus nppiOrC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.4 NppStatus nppiOrC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.5 NppStatus nppiOrC\_16u\_C3IR (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.6 NppStatus nppiOrC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.7 NppStatus nppiOrC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.8 NppStatus nppiOrC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.9 NppStatus nppiOrC\_32s\_AC4IR** (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.10 NppStatus nppiOrC\_32s\_AC4R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.11 NppStatus nppiOrC\_32s\_C1IR** (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.12** `NppStatus nppiOrC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.13** `NppStatus nppiOrC_32s_C3IR (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.14** `NppStatus nppiOrC_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.15 NppStatus nppiOrC\_32s\_C4IR (const Npp32s aConstants[4], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.16 NppStatus nppiOrC\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s aConstants[4], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.17 NppStatus nppiOrC\_8u\_AC4IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.18** `NppStatus nppiOrC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.19** `NppStatus nppiOrC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.20** `NppStatus nppiOrC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.21 NppStatus nppiOrC\_8u\_C3IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.22 NppStatus nppiOrC\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.23 NppStatus nppiOrC\_8u\_C4IR (const Npp8u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.30.2.24 NppStatus nppiOrC\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.31 XorC

Pixel by pixel logical exclusive or of an image with a constant.

### Functions

- **NppStatus nppiXorC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical exclusive or with constant.*

- `NppStatus nppiXorC_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C3IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_32s_AC4IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.*
- `NppStatus nppiXorC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant.*
- `NppStatus nppiXorC_32s_C4IR` (const `Npp32s` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant.*

### 7.31.1 Detailed Description

Pixel by pixel logical exclusive or of an image with a constant.

### 7.31.2 Function Documentation

#### 7.31.2.1 NppStatus nppiXorC\_16u\_AC4IR (const Npp16u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.

##### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.31.2.2 NppStatus nppiXorC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.31.2.3 NppStatus nppiXorC\_16u\_C1IR (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image logical exclusive or with constant.

##### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.4 NppStatus nppiXorC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.5 NppStatus nppiXorC\_16u\_C3IR (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.6 NppStatus nppiXorC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.7 NppStatus nppiXorC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.8 NppStatus nppiXorC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.9 `NppStatus nppiXorC_32s_AC4IR (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.10 `NppStatus nppiXorC_32s_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.11 `NppStatus nppiXorC_32s_C1IR (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel in place image logical exclusive or with constant.

#### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.12 NppStatus nppiXorC\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.13 NppStatus nppiXorC\_32s\_C3IR (const Npp32s aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.14 NppStatus nppiXorC\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.15** `NppStatus nppiXorC_32s_C4IR (const Npp32s aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.16** `NppStatus nppiXorC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.17** `NppStatus nppiXorC_8u_AC4IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.18** `NppStatus nppiXorC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.19** `NppStatus nppiXorC_8u_C1IR (const Npp8u nConstant, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or with constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.20** `NppStatus nppiXorC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.21 NppStatus nppiXorC\_8u\_C3IR (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.22 NppStatus nppiXorC\_8u\_C3R (const Npp8u \* pSrcI, int nSrcIStep, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.23 NppStatus nppiXorC\_8u\_C4IR (const Npp8u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.24 NppStatus nppiXorC\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.32 RShiftC

Pixel by pixel right shift of an image by a constant value.

### Functions

- `NppStatus nppiRShiftC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C1IR` (const `Npp32u` nConstant, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C3IR` (const `Npp32u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8u_AC4IR` (const `Npp32u` aConstants[3], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8u_C4IR` (const `Npp32u` aConstants[4], `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C1R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit signed char channel image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C1IR` (const `Npp32u` nConstant, `Npp8s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit signed char channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_8s_C3R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit signed char channel image right shift by constant.*

- **NppStatus nppiRShiftC\_8s\_C3IR** (const **Npp32u** aConstants[3], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit signed char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_AC4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8s\_AC4IR** (const **Npp32u** aConstants[3], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8s\_C4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_C4IR** (const **Npp32u** aConstants[4], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C1IR** (const **Npp32u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C3IR** (const **Npp32u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_16u\_AC4IR** (const **Npp32u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C4IR** (const **Npp32u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C1R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp32u nConstant`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C1IR` (const `Npp32u nConstant`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C3R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp32u aConstants[3]`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C3IR` (const `Npp32u aConstants[3]`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16s_AC4R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp32u aConstants[3]`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit signed short channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16s_AC4IR` (const `Npp32u aConstants[3]`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16s_C4R` (const `Npp16s *pSrc1`, int `nSrc1Step`, const `Npp32u aConstants[4]`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C4IR` (const `Npp32u aConstants[4]`, `Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C1R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32u nConstant`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*One 32-bit signed integer channel image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C1IR` (const `Npp32u nConstant`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*One 32-bit signed integer channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C3R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32u aConstants[3]`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit signed integer channel image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C3IR` (const `Npp32u aConstants[3]`, `Npp32s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 32-bit signed integer channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_AC4R` (const `Npp32s *pSrc1`, int `nSrc1Step`, const `Npp32u aConstants[3]`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

- `NppStatus nppiRShiftC_32s_AC4IR` (const `Npp32u` `aConstants`[3], `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

- `NppStatus nppiRShiftC_32s_C4R` (const `Npp32s` `*pSrc1`, int `nSrc1Step`, const `Npp32u` `aConstants`[4], `Npp32s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)

Four 32-bit signed integer channel image right shift by constant.

- `NppStatus nppiRShiftC_32s_C4IR` (const `Npp32u` `aConstants`[4], `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)

Four 32-bit signed integer channel in place image right shift by constant.

### 7.32.1 Detailed Description

Pixel by pixel right shift of an image by a constant value.

### 7.32.2 Function Documentation

#### 7.32.2.1 `NppStatus nppiRShiftC_16s_AC4IR` (const `Npp32u` `aConstants`[3], `Npp16s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)

Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.

##### Parameters:

`aConstants` fixed size array of constant values, one per channel.

`pSrcDst` In-Place Image Pointer.

`nSrcDstStep` In-Place-Image Line Step.

`oSizeROI` Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.32.2.2 `NppStatus nppiRShiftC_16s_AC4R` (const `Npp16s` `*pSrc1`, int `nSrc1Step`, const `Npp32u` `aConstants`[3], `Npp16s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)

Four 16-bit signed short channel image right shift by constant with unmodified alpha.

##### Parameters:

`pSrc1` Source-Image Pointer.

`nSrc1Step` Source-Image Line Step.

`aConstants` fixed size array of constant values, one per channel.

`pDst` Destination-Image Pointer.

`nDstStep` Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.3 NppStatus nppiRShiftC\_16s\_C1IR (const Npp32u nConstant, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.4 NppStatus nppiRShiftC\_16s\_C1R (const Npp16s \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.5 NppStatus nppiRShiftC\_16s\_C3IR (const Npp32u aConstants[3], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.6 NppStatus nppiRShiftC\_16s\_C3R (const Npp16s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.7 NppStatus nppiRShiftC\_16s\_C4IR (const Npp32u aConstants[4], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.8 NppStatus nppiRShiftC\_16s\_C4R (const Npp16s \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.9 NppStatus nppiRShiftC\_16u\_AC4IR (const Npp32u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.10 NppStatus nppiRShiftC\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.11 NppStatus nppiRShiftC\_16u\_C1IR (const Npp32u nConstant, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.12 NppStatus nppiRShiftC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.13 NppStatus nppiRShiftC\_16u\_C3IR (const Npp32u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.14** `NppStatus nppiRShiftC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.15** `NppStatus nppiRShiftC_16u_C4IR (const Npp32u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.16** `NppStatus nppiRShiftC_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.17 NppStatus nppiRShiftC\_32s\_AC4IR (const Npp32u aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.18 NppStatus nppiRShiftC\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.19 NppStatus nppiRShiftC\_32s\_C1IR (const Npp32u nConstant, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.20** `NppStatus nppiRShiftC_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.21** `NppStatus nppiRShiftC_32s_C3IR (const Npp32u aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.22** `NppStatus nppiRShiftC_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.23 NppStatus nppiRShiftC\_32s\_C4IR (const Npp32u aConstants[4], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.24 NppStatus nppiRShiftC\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.25 NppStatus nppiRShiftC\_8s\_AC4IR (const Npp32u aConstants[3], Npp8s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.26** `NppStatus nppiRShiftC_8s_AC4R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel image right shift by constant with unmodified alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.27** `NppStatus nppiRShiftC_8s_C1IR (const Npp32u nConstant, Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit signed char channel in place image right shift by constant.

**Parameters:**

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.28** `NppStatus nppiRShiftC_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit signed char channel image right shift by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nConstant* Constant.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.29 NppStatus nppiRShiftC\_8s\_C3IR (const Npp32u aConstants[3], Npp8s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit signed char channel in place image right shift by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.30 NppStatus nppiRShiftC\_8s\_C3R (const Npp8s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit signed char channel image right shift by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.31 NppStatus nppiRShiftC\_8s\_C4IR (const Npp32u aConstants[4], Npp8s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit signed char channel in place image right shift by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.32** `NppStatus nppiRShiftC_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.33** `NppStatus nppiRShiftC_8u_AC4IR (const Npp32u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.34** `NppStatus nppiRShiftC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.35 NppStatus nppiRShiftC\_8u\_C1IR (const Npp32u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.36 NppStatus nppiRShiftC\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.37 NppStatus nppiRShiftC\_8u\_C3IR (const Npp32u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.38** `NppStatus nppiRShiftC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.39** `NppStatus nppiRShiftC_8u_C4IR (const Npp32u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.40** `NppStatus nppiRShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.33 LShiftC

Pixel by pixel left shift of an image by a constant value.

### Functions

- **NppStatus nppiLShiftC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C1IR** (const **Npp32u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C3IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.*
- **NppStatus nppiLShiftC\_8u\_AC4IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.*
- **NppStatus nppiLShiftC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C4IR** (const **Npp32u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C1IR** (const **Npp32u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C3IR` (const `Npp32u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_16u_AC4IR` (const `Npp32u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image left shift by constant.*
- `NppStatus nppiLShiftC_16u_C4IR` (const `Npp32u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C1IR` (const `Npp32u` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C3IR` (const `Npp32u` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_32s_AC4IR` (const `Npp32u` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C4IR` (const `Npp32u` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant.*

### 7.33.1 Detailed Description

Pixel by pixel left shift of an image by a constant value.

### 7.33.2 Function Documentation

#### 7.33.2.1 NppStatus nppiLShiftC\_16u\_AC4IR (const Npp32u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.

##### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.33.2.2 NppStatus nppiLShiftC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.33.2.3 NppStatus nppiLShiftC\_16u\_C1IR (const Npp32u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image left shift by constant.

##### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.4 NppStatus nppiLShiftC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.5 NppStatus nppiLShiftC\_16u\_C3IR (const Npp32u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.6 NppStatus nppiLShiftC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.7 NppStatus nppiLShiftC\_16u\_C4IR (const Npp32u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.8 NppStatus nppiLShiftC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.9 NppStatus nppiLShiftC\_32s\_AC4IR (const Npp32u aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.

#### Parameters:

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.10 NppStatus nppiLShiftC\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)

Four 32-bit signed integer channel image left shift by constant with unmodified alpha.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.11 NppStatus nppiLShiftC\_32s\_C1IR (const Npp32u nConstant, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

One 32-bit signed integer channel in place image left shift by constant.

#### Parameters:

- nConstant* Constant.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.12 NppStatus nppiLShiftC\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.13 NppStatus nppiLShiftC\_32s\_C3IR (const Npp32u aConstants[3], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.14 NppStatus nppiLShiftC\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.15 NppStatus nppiLShiftC\_32s\_C4IR (const Npp32u aConstants[4], Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image left shift by constant.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.16 NppStatus nppiLShiftC\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image left shift by constant.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- aConstants* fixed size array of constant values, one per channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.17 NppStatus nppiLShiftC\_8u\_AC4IR (const Npp32u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.

**Parameters:**

- aConstants* fixed size array of constant values, one per channel.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.18 NppStatus nppiLShiftC\_8u\_AC4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.19 NppStatus nppiLShiftC\_8u\_C11R (const Npp32u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image left shift by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.20 NppStatus nppiLShiftC\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, const Npp32u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.21 NppStatus nppiLShiftC\_8u\_C3IR (const Npp32u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.22 NppStatus nppiLShiftC\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.23 NppStatus nppiLShiftC\_8u\_C4IR (const Npp32u aConstants[4], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.24** `NppStatus nppiLShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.34 And

Pixel by pixel logical and of images.

### Functions

- `NppStatus nppiAnd_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image logical and.*
- `NppStatus nppiAnd_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image logical and.*
- `NppStatus nppiAnd_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image logical and.*
- `NppStatus nppiAnd_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and.*
- `NppStatus nppiAnd_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical and with unmodified alpha.*
- `NppStatus nppiAnd_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with unmodified alpha.*
- `NppStatus nppiAnd_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical and.*
- `NppStatus nppiAnd_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and.*
- `NppStatus nppiAnd_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image logical and.*
- `NppStatus nppiAnd_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image logical and.*
- `NppStatus nppiAnd_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image logical and.*

- `NppStatus nppiAnd_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical and.*
- `NppStatus nppiAnd_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and with unmodified alpha.*
- `NppStatus nppiAnd_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with unmodified alpha.*
- `NppStatus nppiAnd_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical and.*
- `NppStatus nppiAnd_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and.*
- `NppStatus nppiAnd_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical and.*
- `NppStatus nppiAnd_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical and.*
- `NppStatus nppiAnd_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical and.*
- `NppStatus nppiAnd_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical and.*
- `NppStatus nppiAnd_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and with unmodified alpha.*
- `NppStatus nppiAnd_32s_AC4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with unmodified alpha.*
- `NppStatus nppiAnd_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical and.*
- `NppStatus nppiAnd_32s_C4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical and.*

### 7.34.1 Detailed Description

Pixel by pixel logical and of images.

### 7.34.2 Function Documentation

#### 7.34.2.1 `NppStatus nppiAnd_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with unmodified alpha.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.2 `NppStatus nppiAnd_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with unmodified alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.3 `NppStatus nppiAnd_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.4 NppStatus nppiAnd\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.5 NppStatus nppiAnd\_16u\_C3IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.6 NppStatus nppiAnd\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.7 NppStatus nppiAnd\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.8 NppStatus nppiAnd\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.9 NppStatus nppiAnd\_32s\_AC4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.10 NppStatus nppiAnd\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.11 NppStatus nppiAnd\_32s\_C1IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.12 NppStatus nppiAnd\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.13 NppStatus nppiAnd\_32s\_C3IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.14 NppStatus nppiAnd\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.15 NppStatus nppiAnd\_32s\_C4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.16 NppStatus nppiAnd\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.17** `NppStatus nppiAnd_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.18** `NppStatus nppiAnd_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.19 NppStatus nppiAnd\_8u\_C1IR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.20 NppStatus nppiAnd\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.21 NppStatus nppiAnd\_8u\_C3IR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.22** `NppStatus nppiAnd_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.23** `NppStatus nppiAnd_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.24** `NppStatus nppiAnd_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.35 Or

Pixel by pixel logical or of images.

### Functions

- `NppStatus nppiOr_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image logical or.*
- `NppStatus nppiOr_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image logical or.*
- `NppStatus nppiOr_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image logical or.*
- `NppStatus nppiOr_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image logical or.*
- `NppStatus nppiOr_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical or with unmodified alpha.*
- `NppStatus nppiOr_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with unmodified alpha.*
- `NppStatus nppiOr_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical or.*
- `NppStatus nppiOr_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or.*
- `NppStatus nppiOr_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image logical or.*
- `NppStatus nppiOr_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image logical or.*
- `NppStatus nppiOr_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image logical or.*

- `NppStatus nppiOr_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical or.*
- `NppStatus nppiOr_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with unmodified alpha.*
- `NppStatus nppiOr_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with unmodified alpha.*
- `NppStatus nppiOr_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or.*
- `NppStatus nppiOr_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or.*
- `NppStatus nppiOr_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical or.*
- `NppStatus nppiOr_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical or.*
- `NppStatus nppiOr_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with unmodified alpha.*
- `NppStatus nppiOr_32s_AC4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with unmodified alpha.*
- `NppStatus nppiOr_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or.*

### 7.35.1 Detailed Description

Pixel by pixel logical or of images.

### 7.35.2 Function Documentation

#### 7.35.2.1 `NppStatus nppiOr_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with unmodified alpha.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.35.2.2 `NppStatus nppiOr_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with unmodified alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.35.2.3 `NppStatus nppiOr_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.35.2.4 NppStatus nppiOr\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.35.2.5 NppStatus nppiOr\_16u\_C3IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.6 NppStatus nppiOr\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.7 NppStatus nppiOr\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.8 NppStatus nppiOr\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.9 NppStatus nppiOr\_32s\_AC4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.10 NppStatus nppiOr\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.11 NppStatus nppiOr\_32s\_C1IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.12 NppStatus nppiOr\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.13 NppStatus nppiOr\_32s\_C3IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.14 NppStatus nppiOr\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.15 NppStatus nppiOr\_32s\_C4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.16 NppStatus nppiOr\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.17** `NppStatus nppiOr_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.18** `NppStatus nppiOr_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.19 NppStatus nppiOr\_8u\_C1IR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.20 NppStatus nppiOr\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.21 NppStatus nppiOr\_8u\_C3IR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.22** `NppStatus nppiOr_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.23** `NppStatus nppiOr_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.24** `NppStatus nppiOr_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.36 Xor

Pixel by pixel logical exclusive or of images.

### Functions

- `NppStatus nppiXor_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.*
- `NppStatus nppiXor_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.*
- `NppStatus nppiXor_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image logical exclusive or.*
- `NppStatus nppiXor_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image logical exclusive or.*
- `NppStatus nppiXor_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image logical exclusive or.*

- **NppStatus nppiXor\_16u\_C3IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical exclusive or.*
- **NppStatus nppiXor\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or.*
- **NppStatus nppiXor\_16u\_C4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C1IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C3R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C3IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_32s\_AC4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or.*

### 7.36.1 Detailed Description

Pixel by pixel logical exclusive or of images.

### 7.36.2 Function Documentation

#### 7.36.2.1 `NppStatus nppiXor_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.36.2.2 `NppStatus nppiXor_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.36.2.3 `NppStatus nppiXor_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.36.2.4 NppStatus nppiXor\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)

One 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.36.2.5 NppStatus nppiXor\_16u\_C3IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Three 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.6 NppStatus nppiXor\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.7 NppStatus nppiXor\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.8 NppStatus nppiXor\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.9 NppStatus nppiXor\_32s\_AC4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.10 NppStatus nppiXor\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.11 NppStatus nppiXor\_32s\_C1IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.12 NppStatus nppiXor\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.13 NppStatus nppiXor\_32s\_C3IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.14 NppStatus nppiXor\_32s\_C3R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.15 NppStatus nppiXor\_32s\_C4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.16 NppStatus nppiXor\_32s\_C4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.17** `NppStatus nppiXor_8u_AC4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.18** `NppStatus nppiXor_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.19** `NppStatus nppiXor_8u_C1IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.20** `NppStatus nppiXor_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.21** `NppStatus nppiXor_8u_C3IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.22** `NppStatus nppiXor_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.23** `NppStatus nppiXor_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.36.2.24** `NppStatus nppiXor_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.37 Not

Pixel by pixel logical not of image.

### Functions

- **NppStatus nppiNot\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical not.*
- **NppStatus nppiNot\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical not.*
- **NppStatus nppiNot\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical not with unmodified alpha.*
- **NppStatus nppiNot\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical not with unmodified alpha.*
- **NppStatus nppiNot\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical not.*

### 7.37.1 Detailed Description

Pixel by pixel logical not of image.

### 7.37.2 Function Documentation

#### 7.37.2.1 **NppStatus nppiNot\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 8-bit unsigned char channel in place image logical not with unmodified alpha.

#### Parameters:

*pSrcDst* **In-Place Image Pointer.**

*nSrcDstStep* **In-Place-Image Line Step.**

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.2** `NppStatus nppiNot_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical not with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.3** `NppStatus nppiNot_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.4** `NppStatus nppiNot_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.5 NppStatus nppiNot\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.6 NppStatus nppiNot\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.7 NppStatus nppiNot\_8u\_C4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.8 NppStatus nppiNot\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.38 Alpha Composition

### Modules

- [AlphaCompC](#)  
*Composite two images using constant alpha values.*
- [AlphaPremulC](#)  
*Premultiplies pixels of an image using a constant alpha value.*
- [AlphaComp](#)  
*Composite two images using alpha opacity values contained in each image.*
- [AlphaPremul](#)  
*Premultiplies image pixels by image alpha opacity values.*

## 7.39 AlphaCompC

Composite two images using constant alpha values.

### Functions

- `NppStatus nppiAlphaCompC_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 8-bit unsigned char channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Three 8-bit unsigned char channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 8-bit unsigned char channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` nAlpha1, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` nAlpha2, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 8-bit unsigned char channel image composition with alpha using constant source alpha.*
- `NppStatus nppiAlphaCompC_8s_C1R` (const `Npp8s` \*pSrc1, int nSrc1Step, `Npp8s` nAlpha1, const `Npp8s` \*pSrc2, int nSrc2Step, `Npp8s` nAlpha2, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 8-bit signed char channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 16-bit unsigned short channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Three 16-bit unsigned short channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 16-bit unsigned short channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` nAlpha1, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` nAlpha2, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 16-bit unsigned short channel image composition with alpha using constant source alpha.*

- `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s *pSrc1`, int `nSrc1Step`, `Npp16s nAlpha1`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s nAlpha2`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppiAlphaOp eAlphaOp`)  
*One 16-bit signed short channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_32u_C1R` (const `Npp32u *pSrc1`, int `nSrc1Step`, `Npp32u nAlpha1`, const `Npp32u *pSrc2`, int `nSrc2Step`, `Npp32u nAlpha2`, `Npp32u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppiAlphaOp eAlphaOp`)  
*One 32-bit unsigned integer channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_32s_C1R` (const `Npp32s *pSrc1`, int `nSrc1Step`, `Npp32s nAlpha1`, const `Npp32s *pSrc2`, int `nSrc2Step`, `Npp32s nAlpha2`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppiAlphaOp eAlphaOp`)  
*One 32-bit signed integer channel image composition using constant alpha.*
- `NppStatus nppiAlphaCompC_32f_C1R` (const `Npp32f *pSrc1`, int `nSrc1Step`, `Npp32f nAlpha1`, const `Npp32f *pSrc2`, int `nSrc2Step`, `Npp32f nAlpha2`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppiAlphaOp eAlphaOp`)  
*One 32-bit floating point channel image composition using constant alpha.*

### 7.39.1 Detailed Description

Composite two images using constant alpha values.

### 7.39.2 Function Documentation

**7.39.2.1** `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s *pSrc1`, int `nSrc1Step`, `Npp16s nAlpha1`, const `Npp16s *pSrc2`, int `nSrc2Step`, `Npp16s nAlpha2`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppiAlphaOp eAlphaOp`)

One 16-bit signed short channel image composition using constant alpha.

#### Parameters:

- `pSrc1` Source-Image Pointer.
- `nSrc1Step` Source-Image Line Step.
- `nAlpha1` Image alpha opacity (0 - max channel pixel value).
- `pSrc2` Source-Image Pointer.
- `nSrc2Step` Source-Image Line Step.
- `nAlpha2` Image alpha opacity (0 - max channel pixel value).
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).
- `eAlphaOp` alpha-blending operation..

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.39.2.2 NppStatus nppiAlphaCompC\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u \* pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.3 NppStatus nppiAlphaCompC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u \* pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

One 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.4 NppStatus nppiAlphaCompC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u \* pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

Three 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.5 NppStatus nppiAlphaCompC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, const Npp16u \* pSrc2, int nSrc2Step, Npp16u nAlpha2, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

Four 16-bit unsigned short channel image composition using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.6 NppStatus nppiAlphaCompC\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, Npp32f nAlpha1, const Npp32f \* pSrc2, int nSrc2Step, Npp32f nAlpha2, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

One 32-bit floating point channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0.0 - 1.0).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0.0 - 1.0).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.7 NppStatus nppiAlphaCompC\_32s\_C1R (const Npp32s \* pSrc1, int nSrc1Step, Npp32s nAlpha1, const Npp32s \* pSrc2, int nSrc2Step, Npp32s nAlpha2, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

One 32-bit signed integer channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.8 NppStatus nppiAlphaCompC\_32u\_C1R (const Npp32u \* pSrc1, int nSrc1Step, Npp32u nAlpha1, const Npp32u \* pSrc2, int nSrc2Step, Npp32u nAlpha2, Npp32u \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

One 32-bit unsigned integer channel image composition using constant alpha.

**Parameters:**

- pSrc1* [Source-Image Pointer](#).
- nSrc1Step* [Source-Image Line Step](#).
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* [Source-Image Pointer](#).
- nSrc2Step* [Source-Image Line Step](#).
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.9 NppStatus nppiAlphaCompC\_8s\_C1R (const Npp8s \* pSrc1, int nSrc1Step, Npp8s nAlpha1, const Npp8s \* pSrc2, int nSrc2Step, Npp8s nAlpha2, Npp8s \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)**

One 8-bit signed char channel image composition using constant alpha.

**Parameters:**

- pSrc1* [Source-Image Pointer](#).
- nSrc1Step* [Source-Image Line Step](#).
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* [Source-Image Pointer](#).
- nSrc2Step* [Source-Image Line Step](#).
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.10** `NppStatus nppiAlphaCompC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.11** `NppStatus nppiAlphaCompC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.12** `NppStatus nppiAlphaCompC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Three 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.13** `NppStatus nppiAlphaCompC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- nAlpha2* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.40 AlphaPremulC

Premultiplies pixels of an image using a constant alpha value.

### Functions

- **NppStatus** `nppiAlphaPremulC_8u_C1R` (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_C1IR` (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_C3R` (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_C3IR` (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_C4R` (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_C4IR` (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_AC4R` (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_8u_AC4IR` (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_16u_C1R` (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_16u_C1IR` (**Npp16u** nAlpha1, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- **NppStatus** `nppiAlphaPremulC_16u_C3R` (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C3IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_C4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_C4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_AC4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_AC4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.*

### 7.40.1 Detailed Description

Premultiplies pixels of an image using a constant alpha value.

### 7.40.2 Function Documentation

#### 7.40.2.1 `NppStatus nppiAlphaPremulC_16u_AC4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

#### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.40.2.2 `NppStatus nppiAlphaPremulC_16u_AC4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.40.2.3 NppStatus nppiAlphaPremulC\_16u\_C1IR (Npp16u nAlpha1, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.40.2.4 NppStatus nppiAlphaPremulC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image premultiplication using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.40.2.5 `NppStatus nppiAlphaPremulC_16u_C3IR (Npp16u nAlpha1, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel in place image premultiplication using constant alpha.

##### Parameters:

- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.6 `NppStatus nppiAlphaPremulC_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, Npp16u nAlpha1, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 16-bit unsigned short channel image premultiplication using constant alpha.

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.7 `NppStatus nppiAlphaPremulC_16u_C4IR (Npp16u nAlpha1, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image premultiplication using constant alpha.

##### Parameters:

- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.8 NppStatus nppiAlphaPremulC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, Npp16u nAlpha1, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image premultiplication using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.9 NppStatus nppiAlphaPremulC\_8u\_AC4IR (Npp8u nAlpha1, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.

**Parameters:**

- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.10 NppStatus nppiAlphaPremulC\_8u\_AC4R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u nAlpha1, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.11 NppStatus nppiAlphaPremulC\_8u\_C1IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.12 NppStatus nppiAlphaPremulC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image premultiplication using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.13 NppStatus nppiAlphaPremulC\_8u\_C3IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.14 NppStatus nppiAlphaPremulC\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u nAlpha1, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 8-bit unsigned char channel image premultiplication using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.15 NppStatus nppiAlphaPremulC\_8u\_C4IR (Npp8u nAlpha1, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image premultiplication using constant alpha.

**Parameters:**

- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.40.2.16 NppStatus nppiAlphaPremulC\_8u\_C4R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u nAlpha1, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image premultiplication using constant alpha.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- nAlpha1* Image alpha opacity (0 - max channel pixel value).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.41 AlphaComp

Composite two images using alpha opacity values contained in each image.

### Functions

- `NppStatus nppiAlphaComp_8u_AC1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_8s_AC1R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_16u_AC1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_16s_AC1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_32u_AC1R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_32u_AC4R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*
- `NppStatus nppiAlphaComp_32s_AC1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)  
*One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*

- **NppStatus nppiAlphaComp\_32s\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus nppiAlphaComp\_32f\_AC1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*
- **NppStatus nppiAlphaComp\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*

### 7.41.1 Detailed Description

Composite two images using alpha opacity values contained in each image.

### 7.41.2 Function Documentation

**7.41.2.1 NppStatus nppiAlphaComp\_16s\_AC1R** (const **Npp16s** \* pSrc1, int nSrc1Step, const **Npp16s** \* pSrc2, int nSrc2Step, **Npp16s** \* pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.2 NppStatus nppiAlphaComp\_16u\_AC1R** (const **Npp16u** \* pSrc1, int nSrc1Step, const **Npp16u** \* pSrc2, int nSrc2Step, **Npp16u** \* pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.3** `NppStatus nppiAlphaComp_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.4** `NppStatus nppiAlphaComp_32f_AC1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.41.2.5 NppStatus nppiAlphaComp\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)

Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.41.2.6 NppStatus nppiAlphaComp\_32s\_AC1R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)

One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.7 NppStatus nppiAlphaComp\_32s\_AC4R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.8 NppStatus nppiAlphaComp\_32u\_AC1R** (const Npp32u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pSrc2*, int *nSrc2Step*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.9 NppStatus nppiAlphaComp\_32u\_AC4R** (const Npp32u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pSrc2*, int *nSrc2Step*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.10 NppStatus nppiAlphaComp\_8s\_AC1R** (const Npp8s \* *pSrc1*, int *nSrc1Step*, const Npp8s \* *pSrc2*, int *nSrc2Step*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.11 NppStatus nppiAlphaComp\_8u\_AC1R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.12** `NppStatus nppiAlphaComp_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.42 AlphaPremul

Premultiplies image pixels by image alpha opacity values.

### Functions

- `NppStatus nppiAlphaPremul_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).*

- `NppStatus nppiAlphaPremul_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*

- `NppStatus nppiAlphaPremul_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).*

- `NppStatus nppiAlphaPremul_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*

### 7.42.1 Detailed Description

Premultiplies image pixels by image alpha opacity values.

### 7.42.2 Function Documentation

#### 7.42.2.1 `NppStatus nppiAlphaPremul_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.42.2.2 **NppStatus nppiAlphaPremul\_16u\_AC4R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.42.2.3 **NppStatus nppiAlphaPremul\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.42.2.4 **NppStatus nppiAlphaPremul\_8u\_AC4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

## 7.43 Color and Sampling Conversion

Routines manipulating an image's color model and sampling format.

### Modules

- [Color Model Conversion](#)  
*Routines for converting between various image color models.*
- [Color Sampling Format Conversion](#)  
*Routines for converting between various image color sampling formats.*
- [Color Gamma Correction](#)  
*Routines for correcting image color gamma.*
- [Complement Color Key](#)  
*Routines for performing complement color key replacement.*
- [Color Processing](#)  
*Routines for performing image color manipulation.*

### 7.43.1 Detailed Description

Routines manipulating an image's color model and sampling format.

## 7.44 Color Model Conversion

Routines for converting between various image color models.

### RGBToYUV

RGB to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```
Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;
```

- **NppStatus nppiRGBToYUV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.*
- **NppStatus nppiRGBToYUV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.*
- **NppStatus nppiRGBToYUV\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.*
- **NppStatus nppiRGBToYUV\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.*
- **NppStatus nppiRGBToYUV\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.*

### YUVToRGB

YUV to RGB color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```
Npp32f nY = (Npp32f) Y;
Npp32f nU = (Npp32f) U - 128.0F;
Npp32f nV = (Npp32f) V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
```

```

    nR = 0.0F;
    if (nR > 255.0F)
        nR = 255.0F;
    Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
    if (nG < 0.0F)
        nG = 0.0F;
    if (nG > 255.0F)
        nG = 255.0F;
    Npp32f nB = nY + 2.032F * nU;
    if (nB < 0.0F)
        nB = 0.0F;
    if (nB > 255.0F)
        nB = 255.0F;

```

- **NppStatus nppiYUVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYUVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.*
- **NppStatus nppiYUVToRGB\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.*
- **NppStatus nppiYUVToRGB\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.*

## RGBToYUV422

RGB to YUV422 color conversion.

- **NppStatus nppiRGBToYUV422\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.*
- **NppStatus nppiRGBToYUV422\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.*
- **NppStatus nppiRGBToYUV422\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.*

## YUV422ToRGB

YUV422 to RGB color conversion.

- `NppStatus nppiYUV422ToRGB_8u_C2C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3AC4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.*

## RGBToYUV420

RGB to YUV420 color conversion.

- `NppStatus nppiRGBToYUV420_8u_P3R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.*
- `NppStatus nppiRGBToYUV420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.*

## YUV420ToRGB

YUV420 to RGB color conversion.

- `NppStatus nppiYUV420ToRGB_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.*
- `NppStatus nppiYUV420ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYUV420ToRGB_8u_P3AC4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.*

## BGRToYUV420

BGR to YUV420 color conversion.

- `NppStatus nppiBGRToYUV420_8u_AC4P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.*

## YUV420ToBGR

YUV420 to BGR color conversion.

- `NppStatus nppiYUV420ToBGR_8u_P3C3R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToYCbCr

RGB to YCbCr color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YCbCr. In the YCbCr model, Y is defined to have a nominal range [16..235], while Cb and Cr are defined to have a range [16..240], with the value of 128 as corresponding to zero.

```
Npp32f nY = 0.257F * R + 0.504F * G + 0.098F * B + 16.0F;
Npp32f nCb = -0.148F * R - 0.291F * G + 0.439F * B + 128.0F;
Npp32f nCr = 0.439F * R - 0.368F * G - 0.071F * B + 128.0F;
```

- `NppStatus nppiRGBToYCbCr_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.*
- `NppStatus nppiRGBToYCbCr_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.*
- `NppStatus nppiRGBToYCbCr_8u_P3R` (const `Npp8u *const pSrc[3]`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.*
- `NppStatus nppiRGBToYCbCr_8u_C3P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion.*
- `NppStatus nppiRGBToYCbCr_8u_AC4P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.*

## YCbCrToRGB

YCbCr to RGB color conversion.

Here is how NPP converts YCbCr to gamma corrected RGB or BGR. The output RGB values are saturated to the range [0..255].

```
Npp32f nY = 1.164F * ((Npp32f)Y - 16.0F);
Npp32f nR = ((Npp32f)Cr - 128.0F);
Npp32f nB = ((Npp32f)Cb - 128.0F);
Npp32f nG = nY - 0.813F * nR - 0.392F * nB;
if (nG > 255.0F)
    nG = 255.0F;
nR = nY + 1.596F * nR;
if (nR > 255.0F)
    nR = 255.0F;
nB = nY + 2.017F * nB;
if (nB > 255.0F)
    nB = 255.0F;
```

- **NppStatus nppiYCbCrToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.*
- **NppStatus nppiYCbCrToRGB\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.*

## YCbCrToBGR

YCbCr to BGR color conversion.

- **NppStatus nppiYCbCrToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.*
- **NppStatus nppiYCbCrToBGR\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

## YCbCrToBGR\_709CSC

YCbCr to BGR\_709CSC color conversion.

- `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.*
- `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C4R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR\_709CSC color conversion with constant alpha.*

## RGBToYCbCr422

RGB to YCbCr422 color conversion.

- `NppStatus nppiRGBToYCbCr422_8u_C3C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.*
- `NppStatus nppiRGBToYCbCr422_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*
- `NppStatus nppiRGBToYCbCr422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.*

## YCbCr422ToRGB

YCbCr422 to RGB color conversion.

- `NppStatus nppiYCbCr422ToRGB_8u_C2C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYCbCr422ToRGB_8u_C2P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.*
- `NppStatus nppiYCbCr422ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*

## RGBToYCrCb422

RGB to YCrCb422 color conversion.

- `NppStatus nppiRGBToYCrCb422_8u_C3C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*
- `NppStatus nppiRGBToYCrCb422_8u_P3C2R` (const `Npp8u *const pSrc[3]`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*

## YCrCb422ToRGB

YCrCb422 to RGB color conversion.

- `NppStatus nppiYCrCb422ToRGB_8u_C2C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYCrCb422ToRGB_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion.*

## BGRToYCbCr422

BGR to YCbCr422 color conversion.

- `NppStatus nppiBGRToYCbCr422_8u_C3C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*
- `NppStatus nppiBGRToYCbCr422_8u_AC4C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*
- `NppStatus nppiBGRToYCbCr422_8u_C3P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*
- `NppStatus nppiBGRToYCbCr422_8u_AC4P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*

## YCbCr422ToBGR

YCbCr422 to BGR color conversion.

- `NppStatus nppiYCbCr422ToBGR_8u_C2C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion.*
- `NppStatus nppiYCbCr422ToBGR_8u_C2C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nAval`)  
*2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*
- `NppStatus nppiYCbCr422ToBGR_8u_P3C3R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToCbYCr422

RGB to CbYCr422 color conversion.

- `NppStatus nppiRGBToCbYCr422_8u_C3C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*
- `NppStatus nppiRGBToCbYCr422Gamma_8u_C3C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*

## CbYCr422ToRGB

CbYCr422 to RGB color conversion.

- `NppStatus nppiCbYCr422ToRGB_8u_C2C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*

## BGRToCbYCr422

BGR to CbYCr422 color conversion.

- `NppStatus nppiBGRToCbYCr422_8u_AC4C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*

## BGRToCbYCr422\_709HDTV

BGR to CbYCr422\_709HDTV color conversion.

- **NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.*
- **NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_AC4C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.*

## CbYCr422ToBGR

CbYCr422 to BGR color conversion.

- **NppStatus nppiCbYCr422ToBGR\_8u\_C2C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.*

## CbYCr422ToBGR\_709HDTV

CbYCr422 to BGR\_709HDTV color conversion.

- **NppStatus nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR\_709HDTV color conversion.*
- **NppStatus nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.*

## RGBToYCbCr420

RGB to YCbCr420 color conversion.

- **NppStatus nppiRGBToYCbCr420\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*

## YCbCr420ToRGB

YCbCr420 to RGB color conversion.

- `NppStatus nppiYCbCr420ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.*

## RGBToYCrCb420

RGB to YCrCb420 color conversion.

- `NppStatus nppiRGBToYCrCb420_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*

## YCrCb420ToRGB

YCrCb420 to RGB color conversion.

- `NppStatus nppiYCrCb420ToRGB_8u_P3C4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)  
*3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.*

## BGRToYCbCr420

BGR to YCbCr420 color conversion.

- `NppStatus nppiBGRToYCbCr420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*
- `NppStatus nppiBGRToYCbCr420_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*

## BGRToYCbCr420\_709CSC

BGR to YCbCr420\_709CSC color conversion.

- `NppStatus nppiBGRToYCbCr420_709CSC_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.*

- **NppStatus nppiBGRToYCbCr420\_709CSC\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.*

## BGRToYCbCr420\_709HDTV

BGR to YCbCr420\_709HDTV color conversion.

- **NppStatus nppiBGRToYCbCr420\_709HDTV\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709HDTV color conversion.*

## BGRToYCrCb420\_709CSC

BGR to YCrCb420\_709CSC color conversion.

- **NppStatus nppiBGRToYCrCb420\_709CSC\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.*

- **NppStatus nppiBGRToYCrCb420\_709CSC\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.*

## YCbCr420ToBGR

YCbCr420 to BGR color conversion.

- **NppStatus nppiYCbCr420ToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.*

- **NppStatus nppiYCbCr420ToBGR\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)

*3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

### YCbCr420ToBGR\_709CSC

YCbCr420\_709CSC to BGR color conversion.

- `NppStatus nppiYCbCr420ToBGR_709CSC_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.*

### YCbCr420ToBGR\_709HDTV

YCbCr420\_709HDTV to BGR color conversion.

- `NppStatus nppiYCbCr420ToBGR_709HDTV_8u_P3C4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)

*3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.*

### BGRToYCrCb420

BGR to YCrCb420 color conversion.

- `NppStatus nppiBGRToYCrCb420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*

- `NppStatus nppiBGRToYCrCb420_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*

### BGRToYCbCr411

BGR to YCbCr411 color conversion.

- `NppStatus nppiBGRToYCbCr411_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.*

- `NppStatus nppiBGRToYCbCr411_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.*

## YCbCr411ToBGR

YCbCr411 to BGR color conversion.

- `NppStatus nppiYCbCr411ToBGR_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.*
- `NppStatus nppiYCbCr411ToBGR_8u_P3C4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)  
*3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

## RGBToXYZ

RGB to XYZ color conversion.

Here is how NPP converts gamma corrected RGB or BGR to XYZ.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
if (nX > 1.0F)
    nX = 1.0F;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
if (nY > 1.0F)
    nY = 1.0F;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
if (nZ > 1.0F)
    nZ = 1.0F;
X = (Npp8u) (nX * 255.0F);
Y = (Npp8u) (nY * 255.0F);
Z = (Npp8u) (nZ * 255.0F);
```

- `NppStatus nppiRGBToXYZ_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.*
- `NppStatus nppiRGBToXYZ_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.*

## XYZToRGB

XYZ to RGB color conversion.

Here is how NPP converts XYZ to gamma corrected RGB or BGR. The code assumes that X, Y, and Z values are in the range [0..1].

```
Npp32f nNormalizedX = (Npp32f)X * 0.003921569F; // / 255.0F
Npp32f nNormalizedY = (Npp32f)Y * 0.003921569F;
```

```

Npp32f nNormalizedZ = (Npp32f)Z * 0.003921569F;
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

- **NppStatus nppiXYZToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiXYZToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## RGBToLUV

RGB to LUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to CIE LUV using the CIE XYZ D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], U component in the range [-134..220], and V component in the range [-140..122]. The code uses `cbtrf()` the 32 bit floating point cube root math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvn (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
// Now calculate LUV from the XYZ value
Npp32f nTemp = nX + 15.0F * nY + 3.0F * nZ;
Npp32f nu = 4.0F * nX / nTemp;
Npp32f nv = 9.0F * nY / nTemp;
Npp32f nL = 116.0F * cbtrf(nY) - 16.0F;
if (nL < 0.0F)
    nL = 0.0F;
if (nL > 100.0F)
    nL = 100.0F;
nTemp = 13.0F * nL;
Npp32f nU = nTemp * (nu - nun);
if (nU < -134.0F)
    nU = -134.0F;
if (nU > 220.0F)

```

```

    nU = 220.0F;
Npp32f nV = nTemp * (nv - nvN);
if (nV < -140.0F)
    nV = -140.0F;
if (nV > 122.0F)
    nV = 122.0F;
L = (Npp8u) (nL * 255.0F * 0.01F); // / 100.0F
U = (Npp8u) ((nU + 134.0F) * 255.0F * 0.0028249F); // / 354.0F
V = (Npp8u) ((nV + 140.0F) * 255.0F * 0.0038168F); // / 262.0F

```

- **NppStatus nppiRGBToLUV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.*

- **NppStatus nppiRGBToLUV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.*

## LUVToRGB

LUV to RGB color conversion.

Here is how NPP converts CIE LUV to gamma corrected RGB or BGR using the CIE XYZ D65 white point with a Y luminance of 1.0. The code uses powf() the 32 bit floating point power math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert normalized LUV back to original CIE LUV range
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nU = ((Npp32f)U * 354.0F * 0.003921569F) - 134.0F;
Npp32f nV = ((Npp32f)V * 262.0F * 0.003921569F) - 140.0F;
// Now convert LUV to CIE XYZ
Npp32f nTemp = 13.0F * nL;
Npp32f nu = nU / nTemp + nun;
Npp32f nv = nV / nTemp + nvN;
Npp32f nNormalizedY;
if (nL > 7.9996248F)
{
    nNormalizedY = (nL + 16.0F) * 0.008621F; // / 116.0F
    nNormalizedY = powf(nNormalizedY, 3.0F);
}
else
{
    nNormalizedY = nL * 0.001107F; // / 903.3F
}
Npp32f nNormalizedX = (-9.0F * nNormalizedY * nu) / ((nu - 4.0F) * nv - nu * nv);
Npp32f nNormalizedZ = (9.0F * nNormalizedY - 15.0F * nv * nNormalizedY - nv * nNormalizedX) / (3.0F * nv);
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
if (nR < 0.0F)
    nR = 0.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;

```

```

if (nG < 0.0F)
    nG = 0.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
if (nB < 0.0F)
    nB = 0.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiLUVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiLUVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## BGRToLab

BGR to Lab color conversion.

This is how NPP converts gamma corrected BGR or RGB to Lab using the CIE Lab D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], a and b component values are in the range [-128..127]. The code uses `cbtrf()` the 32 bit floating point cube root math function.

```

// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
Npp32f nL = cbtrf(nY);
Npp32f nA;
Npp32f nB;
Npp32f nfX = nX * 1.052128F; // / nCIE_LAB_D65_xn;
Npp32f nfY = nY;
Npp32f nfZ = nZ * 0.918482F; // / nCIE_LAB_D65_zn;
if (nfY > 0.008856F)
{
    nfY = nL - 16.0F;
    nL = 116.0F * nL - 16.0F;
}
else
{
    nfY = 7.787F * nY + 16.0F * 0.008621F; // / 116.0F
}
if (nfX > 0.008856F)
{
    nA = cbtrf(nfX) - 16.0F;
}
else
{

```

```

    nA = 7.787F * nfX + 16.0F * 0.008621F; // / 116.0F
}
nA = 500.0F * (nA - nfY);
if (nfZ > 0.008856F)
{
    nB = cbrtf(nfZ) - 16.0F;
}
else
{
    nB = 7.787F * nfZ + 16.0F * 0.008621F; // / 116.0F
}
nB = 200.0F * (nfY - nB);
// Now scale Lab range
nL = nL * 255.0F * 0.01F; // / 100.0F
nA = nA + 128.0F;
nB = nB + 128.0F;
L = (Npp8u)nL;
a = (Npp8u)nA;
b = (Npp8u)nB;

```

- [NppStatus nppiBGRToLab\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.*

## LabToBGR

Lab to BGR color conversion.

This is how NPP converts Lab to gamma corrected BGR or RGB using the CIE Lab D65 white point with a Y luminance of 1.0. The code uses `powf()` the 32 bit floating point power math function.

```

// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert Lab back to original range then to CIE XYZ
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nA = (Npp32f)a - 128.0F;
Npp32f nB = (Npp32f)b - 128.0F;
Npp32f nP = (nL + 16.0F) * 0.008621F; // / 116.0F
Npp32f nNormalizedY = nP * nP * nP; // powf(nP, 3.0F);
Npp32f nNormalizedX = nCIE_LAB_D65_xn * powf((nP + nA * 0.002F), 3.0F); // / 500.0F
Npp32f nNormalizedZ = nCIE_LAB_D65_zn * powf((nP - nB * 0.005F), 3.0F); // / 200.0F
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);

```

- [NppStatus nppiLabToBGR\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToYCC

RGB to PhotoYCC color conversion.

This is how NPP converts gamma corrected BGR or RGB to PhotoYCC. The computed Y, C1, C2 values are then quantized and converted to fit in the range [0..1] before expanding to 8 bits.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nY = 0.299F * nNormalizedR + 0.587F * nNormalizedG + 0.114F * nNormalizedB;
Npp32f nC1 = nNormalizedB - nY;
nC1 = 111.4F * 0.003921569F * nC1 + 156.0F * 0.003921569F;
Npp32f nC2 = nNormalizedR - nY;
nC2 = 135.64F * 0.003921569F * nC2 + 137.0F * 0.003921569F;
nY = 1.0F * 0.713267F * nY; // / 1.402F
Y = (Npp8u)(nY * 255.0F);
C1 = (Npp8u)(nC1 * 255.0F);
C2 = (Npp8u)(nC2 * 255.0F);
```

- **NppStatus nppiRGBToYCC\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.*
- **NppStatus nppiRGBToYCC\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.*

## YCCToRGB

PhotoYCC to RGB color conversion.

This is how NPP converts PhotoYCC to gamma corrected RGB or BGR.

```
Npp32f nNormalizedY = ((Npp32f)Y * 0.003921569F) * 1.3584F; // / 255.0F
Npp32f nNormalizedC1 = (((Npp32f)C1 * 0.003921569F) - 156.0F * 0.003921569F) * 2.2179F;
Npp32f nNormalizedC2 = (((Npp32f)C2 * 0.003921569F) - 137.0F * 0.003921569F) * 1.8215F;
Npp32f nR = nNormalizedY + nNormalizedC2;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = nNormalizedY - 0.194F * nNormalizedC1 - 0.509F * nNormalizedC2;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = nNormalizedY + nNormalizedC1;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```

- **NppStatus nppiYCCToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYCCToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## RGBToHLS

RGB to HLS color conversion.

This is how NPP converts gamma corrected RGB or BGR to HLS. This code uses the `fmaxf()` and `fminf()` 32 bit floating point math functions.

```

Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Lightness
Npp32f nMax = fmaxf(nNormalizedR, nNormalizedG);
           nMax = fmaxf(nMax, nNormalizedB);
Npp32f nMin = fminf(nNormalizedR, nNormalizedG);
           nMin = fminf(nMin, nNormalizedB);
Npp32f nL = (nMax + nMin) * 0.5F;
Npp32f nDivisor = nMax - nMin;
// Saturation
if (nDivisor == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
{
    if (nL > 0.5F)
        nS = nDivisor / (1.0F - (nMax + nMin - 1.0F));
    else
        nS = nDivisor / (nMax + nMin);
}
// Hue
Npp32f nCr = (nMax - nNormalizedR) / nDivisor;
Npp32f nCg = (nMax - nNormalizedG) / nDivisor;
Npp32f nCb = (nMax - nNormalizedB) / nDivisor;
if (nNormalizedR == nMax)
    nH = nCb - nCg;
else if (nNormalizedG == nMax)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nMax)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
L = (Npp8u) (nL * 255.0F);
S = (Npp8u) (nS * 255.0F);

```

- [NppStatus nppiRGBToHLS\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.*

- [NppStatus nppiRGBToHLS\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*

**HLSToRGB**

HLS to RGB color conversion.

This is how NPP converts HLS to gamma corrected RGB or BGR.

```

Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedL = (Npp32f)L * 0.003921569F;
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nM1;
Npp32f nM2;
Npp32f nR;
Npp32f nG;
Npp32f nB;
Npp32f nh = 0.0F;
if (nNormalizedL <= 0.5F)
    nM2 = nNormalizedL * (1.0F + nNormalizedS);
else
    nM2 = nNormalizedL + nNormalizedS - nNormalizedL * nNormalizedS;
nM1 = 2.0F * nNormalizedL - nM2;
if (nNormalizedS == 0.0F)
    nR = nG = nB = nNormalizedL;
else
{
    nh = nNormalizedH + 0.33333F;
    if (nh > 1.0F)
        nh -= 1.0F;
}
Npp32f nMDiff = nM2 - nM1;
if (0.6667F < nh)
    nR = nM1;
else
{
    if (nh < 0.1667F)
        nR = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nR = nM2;
    else
        nR = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nR > 1.0F)
    nR = 1.0F;
nh = nNormalizedH;
if (0.6667F < nh)
    nG = nM1;
else
{
    if (nh < 0.1667F)
        nG = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nG = nM2;
    else
        nG = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nG > 1.0F)
    nG = 1.0F;
nh = nNormalizedH - 0.33333F;
if (nh < 0.0F)
    nh += 1.0F;
if (0.6667F < nh)
    nB = nM1;
else
{
    if (nh < 0.1667F)
        nB = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nB = nM2;
}

```

```

else
    nB = nM1 + nMDiff * (0.6667F - nh) * 6.0F; // / 0.1667F
}
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiHLSToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiHLSToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## BGRToHLS

BGR to HLS color conversion.

- **NppStatus nppiBGRToHLS\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AP4C4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AP4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.*

## HLSToBGR

HLS to BGR color conversion.

- `NppStatus nppiHLSToBGR_8u_C3P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.*
- `NppStatus nppiHLSToBGR_8u_AC4P4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.*
- `NppStatus nppiHLSToBGR_8u_P3R` (const `Npp8u *const pSrc[3]`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.*
- `NppStatus nppiHLSToBGR_8u_AP4R` (const `Npp8u *const pSrc[4]`, int `nSrcStep`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.*
- `NppStatus nppiHLSToBGR_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.*
- `NppStatus nppiHLSToBGR_8u_P3C3R` (const `Npp8u *const pSrc[3]`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.*
- `NppStatus nppiHLSToBGR_8u_AP4C4R` (const `Npp8u *const pSrc[4]`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.*

## RGBToHSV

RGB to HSV color conversion.

This is how NPP converts gamma corrected RGB or BGR to HSV. This code uses the `fmaxf()` and `fminf()` 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
```

```

Npp32f nH;
// Value
Npp32f nV = fmaxf(nNormalizedR, nNormalizedG);
    nV = fmaxf(nV, nNormalizedB);
// Saturation
Npp32f nTemp = fminf(nNormalizedR, nNormalizedG);
    nTemp = fminf(nTemp, nNormalizedB);
Npp32f nDivisor = nV - nTemp;
if (nV == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
    nS = nDivisor / nV;
// Hue:
Npp32f nCr = (nV - nNormalizedR) / nDivisor;
Npp32f nCg = (nV - nNormalizedG) / nDivisor;
Npp32f nCb = (nV - nNormalizedB) / nDivisor;
if (nNormalizedR == nV)
    nH = nCb - nCg;
else if (nNormalizedG == nV)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nV)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
S = (Npp8u) (nS * 255.0F);
V = (Npp8u) (nV * 255.0F);

```

- **NppStatus nppiRGBToHSV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.*
- **NppStatus nppiRGBToHSV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.*

## HSVToRGB

HSV to RGB color conversion.

This is how NPP converts HSV to gamma corrected RGB or BGR. This code uses the floorf() 32 bit floating point math function.

```

Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nNormalizedV = (Npp32f)V * 0.003921569F;
Npp32f nR;
Npp32f nG;
Npp32f nB;
if (nNormalizedS == 0.0F)
{
    nR = nG = nB = nNormalizedV;
}
else
{
    if (nNormalizedH == 1.0F)

```

```

        nNormalizedH = 0.0F;
    else
        nNormalizedH = nNormalizedH * 6.0F; // / 0.1667F
    }
    Npp32f nI = floorf(nNormalizedH);
    Npp32f nF = nNormalizedH - nI;
    Npp32f nM = nNormalizedV * (1.0F - nNormalizedS);
    Npp32f nN = nNormalizedV * (1.0F - nNormalizedS * nF);
    Npp32f nK = nNormalizedV * (1.0F - nNormalizedS * (1.0F - nF));
    if (nI == 0.0F)
        { nR = nNormalizedV; nG = nK; nB = nM; }
    else if (nI == 1.0F)
        { nR = nN; nG = nNormalizedV; nB = nM; }
    else if (nI == 2.0F)
        { nR = nM; nG = nNormalizedV; nB = nK; }
    else if (nI == 3.0F)
        { nR = nM; nG = nN; nB = nNormalizedV; }
    else if (nI == 4.0F)
        { nR = nK; nG = nM; nB = nNormalizedV; }
    else if (nI == 5.0F)
        { nR = nNormalizedV; nG = nM; nB = nN; }
    R = (Npp8u) (nR * 255.0F);
    G = (Npp8u) (nG * 255.0F);
    B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiHSVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiHSVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## 7.44.1 Detailed Description

Routines for converting between various image color models.

## 7.44.2 Function Documentation

### 7.44.2.1 NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_AC4C2R (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.

images.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.2 NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.3 NppStatus nppiBGRToCbYCr422\_8u\_AC4C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.4 NppStatus nppiBGRToHLS\_8u\_AC4P4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.5 NppStatus nppiBGRToHLS\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.6 NppStatus nppiBGRToHLS\_8u\_AP4C4R (const Npp8u \*const pSrc[4], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.7 NppStatus nppiBGRToHLS\_8u\_AP4R (const Npp8u \*const pSrc[4], int nSrcStep, Npp8u \*pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.8 NppStatus nppiBGRToHLS\_8u\_C3P3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.9 NppStatus nppiBGRToHLS\_8u\_P3C3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.10 NppStatus nppiBGRToHLS\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.11 NppStatus nppiBGRToLab\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.12 NppStatus nppiBGRToYCbCr411\_8u\_AC4P3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.13** `NppStatus nppiBGRToYCbCr411_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.14** `NppStatus nppiBGRToYCbCr420_709CSC_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.15** `NppStatus nppiBGRToYCbCr420_709CSC_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.16 NppStatus nppiBGRToYCbCr420\_709HDTV\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709HDTV color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.17 NppStatus nppiBGRToYCbCr420\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.18** `NppStatus nppiBGRToYCbCr420_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.19** `NppStatus nppiBGRToYCbCr422_8u_AC4C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.20** `NppStatus nppiBGRToYCbCr422_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.21 NppStatus nppiBGRToYCbCr422\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.22 NppStatus nppiBGRToYCbCr422\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.23** `NppStatus nppiBGRToYCrCb420_709CSC_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.24** `NppStatus nppiBGRToYCrCb420_709CSC_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.25** `NppStatus nppiBGRToYCrCb420_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.26 NppStatus nppiBGRToYCrCb420\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.27 NppStatus nppiBGRToYUV420\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.28** `NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR\_709HDTV color conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.29** `NppStatus nppiCbYCr422ToBGR_709HDTV_8u_C2C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.30** `NppStatus nppiCbYCr422ToBGR_8u_C2C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.

images.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.31 NppStatus nppiCbYCr422ToRGB\_8u\_C2C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCrC22 to 3 channel 8-bit unsigned packed RGB color conversion.  
images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.32 NppStatus nppiHLSToBGR\_8u\_AC4P4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.33** `NppStatus nppiHLSToBGR_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.34** `NppStatus nppiHLSToBGR_8u_AP4C4R (const Npp8u *const pSrc[4], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.35** `NppStatus nppiHLSToBGR_8u_AP4R (const Npp8u *const pSrc[4], int nSrcStep, Npp8u * pDst[4], int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.36** `NppStatus nppiHLSToBGR_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.37** `NppStatus nppiHLSToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.38** `NppStatus nppiHLSToBGR_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.39 NppStatus nppiHLSToRGB\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.40 NppStatus nppiHLSToRGB\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.41 NppStatus nppiHSVToRGB\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.42 NppStatus nppiHSVToRGB\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.43 NppStatus nppiLabToBGR\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.44 NppStatus nppiLUVToRGB\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.45 NppStatus nppiLUVToRGB\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.46 NppStatus nppiRGBToCbYCr422\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.47 NppStatus nppiRGBToCbYCr422Gamma\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.48 NppStatus nppiRGBToHLS\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.49 NppStatus nppiRGBToHLS\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.50 NppStatus nppiRGBToHSV\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.51 NppStatus nppiRGBToHSV\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.52 NppStatus nppiRGBToLUV\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.53 NppStatus nppiRGBToLUV\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.54 NppStatus nppiRGBToXYZ\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.55 NppStatus nppiRGBToXYZ\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.56** `NppStatus nppiRGBToYCbCr420_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.57** `NppStatus nppiRGBToYCbCr422_8u_C3C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.58** `NppStatus nppiRGBToYCbCr422_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.59** `NppStatus nppiRGBToYCbCr422_8u_P3C2R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.60** `NppStatus nppiRGBToYCbCr_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.61** `NppStatus nppiRGBToYCbCr_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.62 NppStatus nppiRGBToYCbCr\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.63 NppStatus nppiRGBToYCbCr\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.64 NppStatus nppiRGBToYCbCr\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.65 NppStatus nppiRGBToYCC\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.66 NppStatus nppiRGBToYCC\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.67** `NppStatus nppiRGBToYCrCb420_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.68** `NppStatus nppiRGBToYCrCb422_8u_C3C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.69** `NppStatus nppiRGBToYCrCb422_8u_P3C2R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.70** `NppStatus nppiRGBToYUV420_8u_C3P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.71** `NppStatus nppiRGBToYUV420_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.72** `NppStatus nppiRGBToYUV422_8u_C3C2R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.73** `NppStatus nppiRGBToYUV422_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.74** `NppStatus nppiRGBToYUV422_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.  
 images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.75** `NppStatus nppiRGBToYUV_8u_AC4P4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[4], int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.76** `NppStatus nppiRGBToYUV_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.77** `NppStatus nppiRGBToYUV_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.78** `NppStatus nppiRGBToYUV_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.79** `NppStatus nppiRGBToYUV_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.80** `NppStatus nppiXYZToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.81** `NppStatus nppiXYZToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.82** `NppStatus nppiYCbCr411ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.83** `NppStatus nppiYCbCr411ToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.84** `NppStatus nppiYCbCr420ToBGR_709CSC_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.85** `NppStatus nppiYCbCr420ToBGR_709HDTV_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.86** `NppStatus nppiYCbCr420ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.87** `NppStatus nppiYCbCr420ToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.88** `NppStatus nppiYCbCr420ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.89** `NppStatus nppiYCbCr422ToBGR_8u_C2C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.90** `NppStatus nppiYCbCr422ToBGR_8u_C2C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.91** `NppStatus nppiYCbCr422ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.92** `NppStatus nppiYCbCr422ToRGB_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.93** `NppStatus nppiYCbCr422ToRGB_8u_C2P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.94** `NppStatus nppiYCbCr422ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.95** `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.96** `NppStatus nppiYCbCrToBGR_709CSC_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR\_709CSC color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.97** `NppStatus nppiYCbCrToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.98** `NppStatus nppiYCbCrToBGR_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.99** `NppStatus nppiYCbCrToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.100** `NppStatus nppiYCbCrToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.101** `NppStatus nppiYCbCrToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.102 NppStatus nppiYCbCrToRGB\_8u\_P3C4R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)**

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.103 NppStatus nppiYCbCrToRGB\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.104 NppStatus nppiYCCToRGB\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.105 NppStatus nppiYCCToRGB\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.106 NppStatus nppiYCrCb420ToRGB\_8u\_P3C4R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)**

3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.107 NppStatus nppiYCrCb422ToRGB\_8u\_C2C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.108 NppStatus nppiYCrCb422ToRGB\_8u\_C2P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.109 NppStatus nppiYUV420ToBGR\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.110 NppStatus nppiYUV420ToRGB\_8u\_P3AC4R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.111 NppStatus nppiYUV420ToRGB\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.112 NppStatus nppiYUV420ToRGB\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.113 NppStatus nppiYUV422ToRGB\_8u\_C2C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.114 NppStatus nppiYUV422ToRGB\_8u\_P3AC4R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.115 NppStatus nppiYUV422ToRGB\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.116 NppStatus nppiYUV422ToRGB\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.117 NppStatus nppiYUVToRGB\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.118 NppStatus nppiYUVToRGB\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.119 NppStatus nppiYUVToRGB\_8u\_P3C3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.120** `NppStatus nppiYUVToRGB_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.45 Color Sampling Format Conversion

Routines for converting between various image color sampling formats.

### YCbCr420ToYCbCr411

YCbCr420 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCbCr420ToYCbCr411_8u_P3P2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- `NppStatus nppiYCbCr420ToYCbCr411_8u_P2P3R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

### YCbCr422ToYCbCr422

YCbCr422 to YCbCr422 sampling format conversion.

- `NppStatus nppiYCbCr422_8u_C2P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

### YCbCr422ToYCrCb422

YCbCr422 to YCrCb422 sampling format conversion.

- `NppStatus nppiYCbCr422ToYCrCb422_8u_C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*
- `NppStatus nppiYCbCr422ToYCrCb422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*

**YCbCr422ToCbYCr422**

YCbCr422 to CbYCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToCbYCr422\_8u\_C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

**CbYCr422ToYCbCr411**

CbYCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

**YCbCr422ToYCbCr420**

YCbCr422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int nDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_C2P2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

**YCrCb420ToYCbCr422**

YCrCb420 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb420ToYCbCr422\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus** **nppiYCrCb420ToYCbCr422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

## YCbCr422ToYCrCb420

YCbCr422 to YCrCb420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCrCb420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

## YCbCr422ToYCbCr411

YCbCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_C2P2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCrCb422ToYCbCr422

YCrCb422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr422\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

### YCrCb422ToYCbCr420

YCrCb422 to YCbCr420 sampling format conversion.

- `NppStatus nppiYCrCb422ToYCbCr420_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

### YCrCb422ToYCbCr411

YCrCb422 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCrCb422ToYCbCr411_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

### CbYCr422ToYCbCr422

CbYCr422 to YCbCr422 sampling format conversion.

- `NppStatus nppiCbYCr422ToYCbCr422_8u_C2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

- `NppStatus nppiCbYCr422ToYCbCr422_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

### CbYCr422ToYCbCr420

CbYCr422 to YCbCr420 sampling format conversion.

- `NppStatus nppiCbYCr422ToYCbCr420_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

- `NppStatus nppiCbYCr422ToYCbCr420_8u_C2P2R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDstY`, int `nDstYStep`, `Npp8u *pDstCbCr`, int `nDstCbCrStep`, `NppiSize oSizeROI`)

*2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

### **CbYCr422ToYCrCb420**

CbYCr422 to YCrCb420 sampling format conversion.

- **NppStatus nppiCbYCr422ToYCrCb420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

### **YCbCr420ToYCbCr420**

YCbCr420 to YCbCr420 sampling format conversion.

- **NppStatus nppiYCbCr420\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

- **NppStatus nppiYCbCr420\_8u\_P2P3R** (const **Npp8u** \*const pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

### **YCbCr420ToYCbCr422**

YCbCr420 to YCbCr422 sampling format conversion.

- **NppStatus nppiYCbCr420ToYCbCr422\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int nDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus nppiYCbCr420ToYCbCr422\_8u\_P2P3R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus nppiYCbCr420ToYCbCr422\_8u\_P2C2R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

**YCbCr420ToCbYCr422**

YCbCr420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R` (const `Npp8u *pSrcY`, int `nSrcYStep`, const `Npp8u *pSrcCbCr`, int `nSrcCbCrStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

**YCbCr420ToYCrCb420**

YCbCr420 to YCrCb420 sampling format conversion.

- `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R` (const `Npp8u *pSrcY`, int `nSrcYStep`, const `Npp8u *pSrcCbCr`, int `nSrcCbCrStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

**YCrCb420ToCbYCr422**

YCrCb420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCrCb420ToCbYCr422_8u_P3C2R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

**YCrCb420ToYCbCr420**

YCrCb420 to YCbCr420 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDstY`, int `nDstYStep`, `Npp8u *pDstCbCr`, int `nDstCbCrStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

**YCrCb420ToYCbCr411**

YCrCb420 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDstY`, int `nDstYStep`, `Npp8u *pDstCbCr`, int `nDstCbCrStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCbCr411ToYCbCr411

YCbCr411 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCbCr411_8u_P3P2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- `NppStatus nppiYCbCr411_8u_P2P3R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCbCr411ToYCbCr422

YCbCr411 to YCbCr422 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCbCr422_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P2P3R` (const `Npp8u` \*const pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P2C2R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

## YCbCr411ToYCrCb422

YCbCr411 to YCrCb422 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCrCb422_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCrCb422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*

## YCbCr411ToYCbCr420

YCbCr411 to YCbCr420 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCbCr420_8u_P3R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDst[3]`, int `nDstStep[3]`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr420_8u_P3P2R` (const `Npp8u *const pSrc[3]`, int `rSrcStep[3]`, `Npp8u *pDstY`, int `nDstYStep`, `Npp8u *pDstCbCr`, int `nDstCbCrStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr420_8u_P2P3R` (const `Npp8u *pSrcY`, int `nSrcYStep`, const `Npp8u *pSrcCbCr`, int `nSrcCbCrStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

## YCbCr411ToYCrCb420

YCbCr411 to YCrCb420 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCrCb420_8u_P2P3R` (const `Npp8u *pSrcY`, int `nSrcYStep`, const `Npp8u *pSrcCbCr`, int `nSrcCbCrStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

### 7.45.1 Detailed Description

Routines for converting between various image color sampling formats.

### 7.45.2 Function Documentation

#### 7.45.2.1 `NppStatus nppiCbYCr422ToYCbCr411_8u_C2P3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst[3]`, int `rDstStep[3]`, `NppiSize oSizeROI`)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.2 **NppStatus nppiCbYCr422ToYCbCr420\_8u\_C2P2R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.3 **NppStatus nppiCbYCr422ToYCbCr420\_8u\_C2P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.4 NppStatus nppiCbYCr422ToYCbCr422\_8u\_C2P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.5 NppStatus nppiCbYCr422ToYCbCr422\_8u\_C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.6 NppStatus nppiCbYCr422ToYCrCb420\_8u\_C2P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.7 NppStatus nppiYCbCr411\_8u\_P2P3R (const Npp8u \*pSrcY, int nSrcYStep, const Npp8u \*pSrcCbCr, int nSrcCbCrStep, Npp8u \*pDst[3], int rDstStep[3], NppiSize oSizeROI)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.8 NppStatus nppiYCbCr411\_8u\_P3P2R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDstY, int nDstYStep, Npp8u \*pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.9 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P2P3R** (const Npp8u \* *pSrcY*, int *nSrcYStep*, const Npp8u \* *pSrcCbCr*, int *nSrcCbCrStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.10 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3P2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.11 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.12 NppStatus nppiYCbCr411ToYCbCr422\_8u\_P2C2R (const Npp8u \* *pSrcY*, int *nSrcYStep*, const Npp8u \* *pSrcCbCr*, int *nSrcCbCrStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.13 NppStatus nppiYCbCr411ToYCbCr422\_8u\_P2P3R (const Npp8u \*const *pSrcY*, int *nSrcYStep*, const Npp8u \* *pSrcCbCr*, int *nSrcCbCrStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.14** `NppStatus nppiYCbCr411ToYCbCr422_8u_P3C2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.15** `NppStatus nppiYCbCr411ToYCbCr422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Planar-Image Pointer Array.
- nDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.16** `NppStatus nppiYCbCr411ToYCrCb420_8u_P2P3R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

**Parameters:**

- pSrcY* Source-Planar-Image Pointer.
- nSrcYStep* Source-Planar-Image Line Step.

*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.17 **NppStatus nppiYCbCr411ToYCrCb422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.18 **NppStatus nppiYCbCr411ToYCrCb422\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.19** `NppStatus nppiYCbCr420_8u_P2P3R (const Npp8u *const pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.20** `NppStatus nppiYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.21** `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.22** `NppStatus nppiYCbCr420ToYCbCr411_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.23** `NppStatus nppiYCbCr420ToYCbCr411_8u_P3P2R (const Npp8u * const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.24** `NppStatus nppiYCbCr420ToYCbCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.25** `NppStatus nppiYCbCr420ToYCbCr422_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.26** `NppStatus nppiYCbCr420ToYCbCr422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst[3], int nDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Planar-Image Pointer Array.
- rSrcStep* Source-Planar-Image Line Step Array.
- pDst* Destination-Planar-Image Pointer Array.
- nDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.27** `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

**Parameters:**

- pSrcY* Source-Planar-Image Pointer.
- nSrcYStep* Source-Planar-Image Line Step.
- pSrcCbCr* Source-Planar-Image Pointer.
- nSrcCbCrStep* Source-Planar-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.28** `NppStatus nppiYCbCr422_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.29 NppStatus nppiYCbCr422\_8u\_P3C2R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.30 NppStatus nppiYCbCr422ToCbYCr422\_8u\_C2R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.31** `NppStatus nppiYCbCr422ToYCbCr411_8u_C2P2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDstY* Destination-Planar-Image Pointer.
- nDstYStep* Destination-Planar-Image Line Step.
- pDstCbCr* Destination-Planar-Image Pointer.
- nDstCbCrStep* Destination-Planar-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.32** `NppStatus nppiYCbCr422ToYCbCr411_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Planar-Image Pointer Array.
- rDstStep* Destination-Planar-Image Line Step Array.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.33** `NppStatus nppiYCbCr422ToYCbCr411_8u_P3P2R (const Npp8u * const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.34 **NppStatus nppiYCbCr422ToYCbCr411\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.35 **NppStatus nppiYCbCr422ToYCbCr420\_8u\_C2P2R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.36** `NppStatus nppiYCbCr422ToYCbCr420_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.37** `NppStatus nppiYCbCr422ToYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDstY* Destination-Planar-Image Pointer.

*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.38** `NppStatus nppiYCbCr422ToYCbCr420_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u * pDst[3], int nDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.39** `NppStatus nppiYCbCr422ToYCrCb420_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.40** `NppStatus nppiYCbCr422ToYCrCb422_8u_C2R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.41 NppStatus nppiYCbCr422ToYCrCb422\_8u\_P3C2R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.42 NppStatus nppiYCrCb420ToCbYCr422\_8u\_P3C2R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.43** `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDstY* Destination-Planar-Image Pointer.

*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.44** `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDstY* Destination-Planar-Image Pointer.

*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.45** `NppStatus nppiYCrCb420ToYCbCr422_8u_P3C2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.46** `NppStatus nppiYCrCb420ToYCbCr422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.45.2.47** `NppStatus nppiYCrCb422ToYCbCr411_8u_C2P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.48 NppStatus nppiYCrCb422ToYCbCr420\_8u\_C2P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.49 NppStatus nppiYCrCb422ToYCbCr422\_8u\_C2P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.46 Color Gamma Correction

Routines for correcting image color gamma.

### GammaFwd

Forward gamma correction.

- `NppStatus nppiGammaFwd_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed color not in place forward gamma correction.*
- `NppStatus nppiGammaFwd_8u_C3IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed color in place forward gamma correction.*
- `NppStatus nppiGammaFwd_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.*
- `NppStatus nppiGammaFwd_8u_AC4IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.*
- `NppStatus nppiGammaFwd_8u_P3R` (const `Npp8u *const pSrc[3]`, int `nSrcStep`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar color not in place forward gamma correction.*
- `NppStatus nppiGammaFwd_8u_IP3R` (`Npp8u *const pSrcDst[3]`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned planar color in place forward gamma correction.*

### GammaInv

Inverse gamma correction.

- `NppStatus nppiGammaInv_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed color not in place inverse gamma correction.*
- `NppStatus nppiGammaInv_8u_C3IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*3 channel 8-bit unsigned packed color in place inverse gamma correction.*
- `NppStatus nppiGammaInv_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.*
- `NppStatus nppiGammaInv_8u_AC4IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)  
*4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.*

- **NppStatus nppiGammaInv\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar color not in place inverse gamma correction.*
- **NppStatus nppiGammaInv\_8u\_IP3R** (**Npp8u** \*const pSrcDst[3], int nSrcDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar color in place inverse gamma correction.*

### 7.46.1 Detailed Description

Routines for correcting image color gamma.

### 7.46.2 Function Documentation

#### 7.46.2.1 **NppStatus nppiGammaFwd\_8u\_AC4IR** (**Npp8u** \* pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.

##### Parameters:

- pSrcDst* in place packed pixel format image pointer.
- nSrcDstStep* in place packed pixel format image line step.
- oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.46.2.2 **NppStatus nppiGammaFwd\_8u\_AC4R** (const **Npp8u** \* pSrc, int nSrcStep, **Npp8u** \* pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.

##### Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.3 NppStatus nppiGammaFwd\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed color in place forward gamma correction.

**Parameters:**

*pSrcDst* in place packed pixel image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.4 NppStatus nppiGammaFwd\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed color not in place forward gamma correction.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.5 NppStatus nppiGammaFwd\_8u\_IP3R (Npp8u \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar color in place forward gamma correction.

**Parameters:**

*pSrcDst* in place planar pixel format image pointer array.  
*nSrcDstStep* in place planar pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.6 NppStatus nppiGammaFwd\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar color not in place forward gamma correction.

**Parameters:**

*pSrc* source planar pixel format image pointer array.  
*nSrcStep* source planar pixel format image line step.  
*pDst* destination planar pixel format image pointer array.  
*nDstStep* destination planar pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.7 NppStatus nppiGammaInv\_8u\_AC4IR (Npp8u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.8 NppStatus nppiGammaInv\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.9 NppStatus nppiGammaInv\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color in place inverse gamma correction.

#### Parameters:

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.10 NppStatus nppiGammaInv\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color not in place inverse gamma correction.

#### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.11 NppStatus nppiGammaInv\_8u\_IP3R (Npp8u \*const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar color in place inverse gamma correction.

#### Parameters:

*pSrcDst* in place planar pixel format image pointer array.  
*nSrcDstStep* in place planar pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.12** `NppStatus nppiGammaInv_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar color not in place inverse gamma correction.

**Parameters:**

*pSrc* source planar pixel format image pointer array.

*nSrcStep* source planar pixel format image line step.

*pDst* destination planar pixel format image pointer array.

*nDstStep* destination planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.47 Complement Color Key

Routines for performing complement color key replacement.

### CompColorKey

Complement color key replacement.

- `NppStatus nppiCompColorKey_8u_C1R` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nColorKeyConst`)  
*1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- `NppStatus nppiCompColorKey_8u_C3R` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nColorKeyConst[3]`)  
*3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- `NppStatus nppiCompColorKey_8u_C4R` (const `Npp8u *pSrc1`, int `nSrc1Step`, const `Npp8u *pSrc2`, int `nSrc2Step`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nColorKeyConst[4]`)  
*4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- `NppStatus nppiAlphaCompColorKey_8u_AC4R` (const `Npp8u *pSrc1`, int `nSrc1Step`, `Npp8u nAlpha1`, const `Npp8u *pSrc2`, int `nSrc2Step`, `Npp8u nAlpha2`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nColorKeyConst[4]`, `NppiAlphaOp nppAlphaOp`)  
*4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.*

### 7.47.1 Detailed Description

Routines for performing complement color key replacement.

### 7.47.2 Function Documentation

- 7.47.2.1 `NppStatus nppiAlphaCompColorKey_8u_AC4R` (const `Npp8u *pSrc1`, int `nSrc1Step`, `Npp8u nAlpha1`, const `Npp8u *pSrc2`, int `nSrc2Step`, `Npp8u nAlpha2`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, `Npp8u nColorKeyConst[4]`, `NppiAlphaOp nppAlphaOp`)**

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.

#### Parameters:

- `pSrc1`* source1 packed pixel format image pointer.
- `nSrc1Step`* source1 packed pixel format image line step.
- `nAlpha1`* source1 image alpha opacity (0 - max channel pixel value).
- `pSrc2`* source2 packed pixel format image pointer.

*nSrc2Step* source2 packed pixel format image line step.  
*nAlpha2* source2 image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nColorKeyConst* color key constant array  
*nppAlphaOp* NppiAlphaOp alpha compositing operation selector (excluding premul ops).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.47.2.2 NppStatus nppiCompColorKey\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nColorKeyConst)

1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nColorKeyConst* color key constant

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.47.2.3 NppStatus nppiCompColorKey\_8u\_C3R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nColorKeyConst[3])

3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nColorKeyConst* color key constant array

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.47.2.4** `NppStatus nppiCompColorKey_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nColorKeyConst[4])`

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nColorKeyConst* color key constant array

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.48 Color Processing

Routines for performing image color manipulation.

### ColorTwist

Perform color twist pixel processing.

- `NppStatus nppiColorTwist32f_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*3 channel 8-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_8u_P3R` (const `Npp8u` \*const \*pSrc, int nSrcStep, `Npp8u` \*\*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*3 channel planar 8-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` twist[3][4])  
*4 channel 8-bit unsigned color twist, not affecting Alpha.*

### ColorLUT

Perform image color processing using various types of color look up tables.

- `NppStatus nppiLUT_Linear_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.*

#### 7.48.1 Detailed Description

Routines for performing image color manipulation.

## 7.48.2 Function Documentation

### 7.48.2.1 `NppStatus nppiColorTwist32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f twist[3][4])`

4 channel 8-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point pixel values is applied with in ROI. Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*twist* The color twist matrix with floating-point pixel values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.2 `NppStatus nppiColorTwist32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f twist[3][4])`

3 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point pixel values is applied within ROI.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*twist* The color twist matrix with floating-point pixel values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.3 `NppStatus nppiColorTwist32f_8u_P3R (const Npp8u *const * pSrc, int nSrcStep, Npp8u ** pDst, int nDstStep, NppiSize oSizeROI, const Npp32f twist[3][4])`

3 channel planar 8-bit unsigned color twist.

An input color twist matrix with floating-point pixel values is applied within ROI.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- twist* The color twist matrix with floating-point pixel values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.48.2.4 **NppStatus nppiLUT\_Linear\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues[3], const Npp32s \* pLevels[3], int nLevels[3])**

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

>>>>>> ATTENTION ATTENTION <<<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<<

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.
- pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.
- nLevels* Host pointer to an array of 3 user defined input/output mapping points, one per color CHANNEL.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR** if the number of levels is less than 2.

#### 7.48.2.5 NppStatus nppiLUT\_Linear\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues, const Npp32s \* pLevels, int nLevels)

8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be device memory pointers.

>>>>>> <<<<<<<<

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Pointer to an array of user defined OUTPUT values (this is now a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is now a device memory pointer)

*nLevels* Number of user defined input/output mapping points (levels)

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2.

#### 7.48.2.6 NppStatus nppiLUT\_Linear\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues[3], const Npp32s \* pLevels[3], int nLevels[3])

3 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<<

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined input/output mapping points, one per color CHANNEL.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2.

**7.48.2.7 NppStatus nppiLUT\_Linear\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues[4], const Npp32s \* pLevels[4], int nLevels[4])**

4 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<<

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined input/output mapping points, one per color CHANNEL.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2.

## 7.49 Compression

Image compression primitives.

### Modules

- [Quantization Functions](#)

### 7.49.1 Detailed Description

Image compression primitives.

The JPEG standard defines a flow of level shift, DCT and quantization for forward JPEG transform and inverse level shift, IDCT and de-quantization for inverse JPEG transform. This group has the functions for both forward and inverse functions.

## 7.50 Quantization Functions

### Functions

- **NppStatus** `nppiQuantFwdRawTableInit_JPEG_8u` (**Npp8u** \*hpQuantRawTable, int nQualityFactor)  
*Apply quality factor to raw 8-bit quantization table.*
- **NppStatus** `nppiQuantFwdTableInit_JPEG_8u16u` (const **Npp8u** \*hpQuantRawTable, **Npp16u** \*hpQuantFwdRawTable)  
*Initializes a quantization table for `nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R()`.*
- **NppStatus** `nppiQuantInvTableInit_JPEG_8u16u` (const **Npp8u** \*hpQuantRawTable, **Npp16u** \*hpQuantFwdRawTable)  
*Initializes a quantization table for `nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R()`.*
- **NppStatus** `nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R` (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, const **Npp16u** \*pQuantFwdTable, **NppiSize** oSizeROI)  
*Forward DCT, quantization and level shift part of the JPEG encoding.*
- **NppStatus** `nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R` (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, const **Npp16u** \*pQuantInvTable, **NppiSize** oSizeROI)  
*Inverse DCT, de-quantization and level shift part of the JPEG decoding.*

### 7.50.1 Function Documentation

#### 7.50.1.1 **NppStatus** `nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R` (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, const **Npp16u** \*pQuantFwdTable, **NppiSize** oSizeROI)

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pQuantFwdTable* Forward quantization tables for JPEG encoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Error codes:

- **NPP\_SIZE\_ERROR** For negative input height/width or not a multiple of 8 width/height.
- **NPP\_STEP\_ERROR** If input image width is not multiple of 8 or does not match ROI.
- **NPP\_NULL\_POINTER\_ERROR** If the destination pointer is 0.

**7.50.1.2 NppStatus nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, const Npp16u \* pQuantInvTable, NppiSize oSizeROI)**

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pQuantInvTable* Inverse quantization tables for JPEG decoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) For negative input height/width or not a multiple of 8 width/height.
- [NPP\\_STEP\\_ERROR](#) If input image width is not multiple of 8 or does not match ROI.
- [NPP\\_NULL\\_POINTER\\_ERROR](#) If the destination pointer is 0.

**7.50.1.3 NppStatus nppiQuantFwdRawTableInit\_JPEG\_8u (Npp8u \* hpQuantRawTable, int nQualityFactor)**

Apply quality factor to raw 8-bit quantization table.

This is effectively an in-place method that modifies a given raw quantization table based on a quality factor. Note that this method is a host method and that the pointer to the raw quantization table is a host pointer.

**Parameters:**

*hpQuantRawTable* Raw quantization table.

*nQualityFactor* Quality factor for the table. Range is [1:100].

**Returns:**

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if *hpQuantRawTable* is 0.

**7.50.1.4 NppStatus nppiQuantFwdTableInit\_JPEG\_8u16u (const Npp8u \* hpQuantRawTable, Npp16u \* hpQuantFwdRawTable)**

Initializes a quantization table for `nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R()`.

The method creates a 16-bit version of the raw table and converts the data order from zigzag layout to original row-order layout since raw quantization tables are typically stored in zigzag format.

This method is a host method. It consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with `nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R()` function.

**Parameters:**

*hpQuantRawTable* Host pointer to raw quantization table as returned by [nppiQuantFwdRawTableInit\\_JPEG\\_8u\(\)](#). The raw quantization table is assumed to be in zigzag order.

*hpQuantFwdRawTable* Forward quantization table for use with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).

**Returns:**

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) pQuantRawTable is 0.

**7.50.1.5 NppStatus nppiQuantInvTableInit\_JPEG\_8u16u (const Npp8u \* hpQuantRawTable, Npp16u \* hpQuantFwdRawTable)**

Initializes a quantization table for [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\(\)](#).

The [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) method uses a quantization table in a 16-bit format allowing for faster processing. In addition it converts the data order from zigzag layout to original row-order layout. Typically raw quantization tables are stored in zigzag format.

This method is a host method and consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) function.

**Parameters:**

*hpQuantRawTable* Raw quantization table.

*hpQuantFwdRawTable* Inverse quantization table.

**Returns:**

[NPP\\_NULL\\_POINTER\\_ERROR](#) pQuantRawTable or pQuantFwdRawTable is 0.

## 7.51 Labeling and Segmentation

Pixel labeling and image segmentation operations.

### Modules

- [GraphCut](#)

### Typedefs

- typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

#### 7.51.1 Detailed Description

Pixel labeling and image segmentation operations.

#### 7.51.2 Typedef Documentation

##### 7.51.2.1 typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

## 7.52 GraphCut

### Graphcut

- [NppStatus nppiGraphcutGetSize](#) ([NppiSize](#) oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.*
- [NppStatus nppiGraphcut8GetSize](#) ([NppiSize](#) oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcutInitAlloc](#) ([NppiSize](#) oSize, [NppiGraphcutState](#) \*\*ppState, [Npp8u](#) \*pDeviceMem)  
*Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcut8InitAlloc](#) ([NppiSize](#) oSize, [NppiGraphcutState](#) \*\*ppState, [Npp8u](#) \*pDeviceMem)  
*Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcutFree](#) ([NppiGraphcutState](#) \*pState)  
*Frees the additional resources of the graph-cut state structure.*
- [NppStatus nppiGraphcut\\_32s8u](#) ([Npp32s](#) \*pTerminals, [Npp32s](#) \*pLeftTransposed, [Npp32s](#) \*pRightTransposed, [Npp32s](#) \*pTop, [Npp32s](#) \*pBottom, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*
- [NppStatus nppiGraphcut8\\_32s8u](#) ([Npp32s](#) \*pTerminals, [Npp32s](#) \*pLeftTransposed, [Npp32s](#) \*pRightTransposed, [Npp32s](#) \*pTop, [Npp32s](#) \*pTopLeft, [Npp32s](#) \*pTopRight, [Npp32s](#) \*pBottom, [Npp32s](#) \*pBottomLeft, [Npp32s](#) \*pBottomRight, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*
- [NppStatus nppiGraphcut\\_32f8u](#) ([Npp32f](#) \*pTerminals, [Npp32f](#) \*pLeftTransposed, [Npp32f](#) \*pRightTransposed, [Npp32f](#) \*pTop, [Npp32f](#) \*pBottom, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit float edge capacities).*
- [NppStatus nppiGraphcut8\\_32f8u](#) ([Npp32f](#) \*pTerminals, [Npp32f](#) \*pLeftTransposed, [Npp32f](#) \*pRightTransposed, [Npp32f](#) \*pTop, [Npp32f](#) \*pTopLeft, [Npp32f](#) \*pTopRight, [Npp32f](#) \*pBottom, [Npp32f](#) \*pBottomLeft, [Npp32f](#) \*pBottomRight, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit float edge capacities).*

## 7.52.1 Function Documentation

**7.52.1.1** `NppStatus nppiGraphcut8_32f8u (Npp32f * pTerminals, Npp32f * pLeftTransposed, Npp32f * pRightTransposed, Npp32f * pTop, Npp32f * pTopLeft, Npp32f * pTopRight, Npp32f * pBottom, Npp32f * pBottomLeft, Npp32f * pBottomRight, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( `terminals(x) = source(x) - sink(x)` ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and  $>0$ ).

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

- pTerminals* Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)
- pLeftTransposed* Pointer to transposed left edge capacities (`left(0,*)` must be 0)
- pRightTransposed* Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)
- pTop* Pointer to top edge capacities (`top(*,0)` must be 0)
- pTopLeft* Pointer to top left edge capacities (`topleft(*,0)` & `topleft(0,*)` must be 0)
- pTopRight* Pointer to top right edge capacities (`topright(*,0)` & `topright(width-1,*)` must be 0)
- pBottom* Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)
- pBottomLeft* Pointer to bottom left edge capacities (`bottomleft(*,height-1)` && `bottomleft(0,*)` must be 0)
- pBottomRight* Pointer to bottom right edge capacities (`bottomright(*,height-1)` && `bottomright(width-1,*)` must be 0)
- nStep* Step in bytes between any pair of sequential rows of edge capacities
- nTransposedStep* Step in bytes between any pair of sequential rows of transposed edge capacities
- size* Graph size
- pLabel* Pointer to destination label image
- nLabelStep* Step in bytes between any pair of sequential rows of label image
- pState* Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.2 NppStatus nppiGraphcut8\_32s8u (Npp32s \* pTerminals, Npp32s \* pLeftTransposed, Npp32s \* pRightTransposed, Npp32s \* pTop, Npp32s \* pTopLeft, Npp32s \* pTopRight, Npp32s \* pBottom, Npp32s \* pBottomLeft, Npp32s \* pBottomRight, int nStep, int nTransposedStep, NppiSize size, Npp8u \* pLabel, int nLabelStep, NppiGraphcutState \* pState)**

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (  $terminals(x) = source(x) - sink(x)$  ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example  $left(0,*) == 0$ ). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcut8Free\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

**pTerminals** Pointer to differences of terminal edge capacities ( $terminal(x) = source(x) - sink(x)$ )

**pLeftTransposed** Pointer to transposed left edge capacities ( $left(0,*)$  must be 0)

**pRightTransposed** Pointer to transposed right edge capacities ( $right(width-1,*)$  must be 0)

**pTop** Pointer to top edge capacities ( $top(*,0)$  must be 0)

**pTopLeft** Pointer to top left edge capacities ( $opleft(*,0)$  &  $opleft(0,*)$  must be 0)

**pTopRight** Pointer to top right edge capacities ( $topright(*,0)$  &  $topright(width-1,*)$  must be 0)

**pBottom** Pointer to bottom edge capacities ( $bottom(*,height-1)$  must be 0)

**pBottomLeft** Pointer to bottom left edge capacities ( $bottomleft(*,height-1)$  &&  $bottomleft(0,*)$  must be 0)

**pBottomRight** Pointer to bottom right edge capacities ( $bottomright(*,height-1)$  &&  $bottomright(width-1,*)$  must be 0)

**nStep** Step in bytes between any pair of sequential rows of edge capacities

**nTransposedStep** Step in bytes between any pair of sequential rows of tranposed edge capacities

**size** Graph size

**pLabel** Pointer to destination label image

**nLabelStep** Step in bytes between any pair of sequential rows of label image

**pState** Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.3 NppStatus nppiGraphcut8GetSize (NppiSize oSize, int \* pBufSize)**

Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcut8\\_32s8u\(\)](#).

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

#### 7.52.1.4 NppStatus nppiGraphcut8InitAlloc (NppiSize *oSize*, NppiGraphcutState \*\* *ppState*, Npp8u \* *pDeviceMem*)

Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8\\_32s8u\(\)](#), [nppiGraphcut8GetSize\(\)](#).

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcut8GetSize\(\)](#).

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

#### 7.52.1.5 NppStatus nppiGraphcut\_32f8u (Npp32f \* *pTerminals*, Npp32f \* *pLeftTransposed*, Npp32f \* *pRightTransposed*, Npp32f \* *pTop*, Npp32f \* *pBottom*, int *nStep*, int *nTransposedStep*, NppiSize *size*, Npp8u \* *pLabel*, int *nLabelStep*, NppiGraphcutState \* *pState*)

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array (  $terminals(x) = source(x) - sink(x)$  ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example  $left(0,*) == 0$ ). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

**See also:**

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

**Parameters:**

*pTerminals* Pointer to differences of terminal edge capacities (terminal(x) = source(x) - sink(x))  
*pLeftTransposed* Pointer to transposed left edge capacities (left(0,\*) must be 0)  
*pRightTransposed* Pointer to transposed right edge capacities (right(width-1,\*) must be 0)  
*pTop* Pointer to top edge capacities (top(\*,0) must be 0)  
*pBottom* Pointer to bottom edge capacities (bottom(\*,height-1) must be 0)  
*nStep* Step in bytes between any pair of sequential rows of edge capacities  
*nTransposedStep* Step in bytes between any pair of sequential rows of transposed edge capacities  
*size* Graph size  
*pLabel* Pointer to destination label image  
*nLabelStep* Step in bytes between any pair of sequential rows of label image  
*pState* Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.6** `NppStatus nppiGraphcut_32s8u (Npp32s * pTerminals, Npp32s * pLeftTransposed, Npp32s * pRightTransposed, Npp32s * pTop, Npp32s * pBottom, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( terminals(x) = source(x) - sink(x) ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example left(0,\*) == 0). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

**See also:**

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

**Parameters:**

*pTerminals* Pointer to differences of terminal edge capacities (terminal(x) = source(x) - sink(x))  
*pLeftTransposed* Pointer to transposed left edge capacities (left(0,\*) must be 0)  
*pRightTransposed* Pointer to transposed right edge capacities (right(width-1,\*) must be 0)  
*pTop* Pointer to top edge capacities (top(\*,0) must be 0)  
*pBottom* Pointer to bottom edge capacities (bottom(\*,height-1) must be 0)  
*nStep* Step in bytes between any pair of sequential rows of edge capacities

*nTransposedStep* Step in bytes between any pair of sequential rows of tranposed edge capacities

*size* Graph size

*pLabel* Pointer to destination label image

*nLabelStep* Step in bytes between any pair of sequential rows of label image

*pState* Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.7 NppStatus nppiGraphcutFree (NppiGraphcutState \* pState)**

Frees the additional resources of the graph-cut state structure.

**See also:**

[nppiGraphcutInitAlloc](#)  
[nppiGraphcut8InitAlloc](#)

**Parameters:**

*pState* Pointer to graph-cut state structure.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pState pointer is NULL

**7.52.1.8 NppStatus nppiGraphcutGetSize (NppiSize oSize, int \* pBufSize)**

Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.

**See also:**

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcut\\_32s8u\(\)](#).

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.52.1.9 NppStatus nppiGraphcutInitAlloc (NppiSize *oSize*, NppiGraphcutState \*\* *ppState*, Npp8u \* *pDeviceMem*)**

Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.

**See also:**

[nppiGraphcut\\_32s8u\(\)](#), [nppiGraphcutGetSize\(\)](#).

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* pDeviceMem to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcutGetSize\(\)](#).

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

## 7.53 Data Exchange and Initialization

Primitives for initialization, copying and converting image data.

### Modules

- [Set](#)
- [Copy](#)
- [Convert](#)
- [Copy Constant Border](#)
- [Transpose And Swap Channels](#)

### 7.53.1 Detailed Description

Primitives for initialization, copying and converting image data.

## 7.54 Set

### Image-Memory Set

Set methods for images of various types.

Images are passed to these primitives via a pointer to the image data (first pixel in the ROI) and a step-width, i.e. the number of bytes between successive lines. The size of the area to be set (region-of-interest, ROI) is specified via a [NppiSize](#) struct. In addition to the image data and ROI, all methods have a parameter to specify the value being set. In case of single channel images this is a single value, in case of multi-channel, an array of values is passed.

- [NppStatus nppiSet\\_8s\\_C1R](#) ([Npp8s](#) nValue, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit image set.*
- [NppStatus nppiSet\\_8s\\_C2R](#) ([Npp8s](#) aValue[2], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit two-channel image set.*
- [NppStatus nppiSet\\_8s\\_C3R](#) ([Npp8s](#) aValue[3], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit three-channel image set.*
- [NppStatus nppiSet\\_8s\\_C4R](#) ([Npp8s](#) aValue[4], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit four-channel image set.*
- [NppStatus nppiSet\\_8s\\_AC4R](#) ([Npp8s](#) aValue[3], [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit four-channel image set ignoring alpha channel.*
- [NppStatus nppiSet\\_8u\\_C1R](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C1MR](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C4R](#) (const [Npp8u](#) aValues[4], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_C4MR](#) (const [Npp8u](#) aValues[4], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 8-bit unsigned image set.*
- [NppStatus nppiSet\\_8u\\_AC4R](#) (const [Npp8u](#) aValues[3], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 8-bit unsigned image set method, not affecting Alpha channel.*
- [NppStatus nppiSet\\_8u\\_AC4MR](#) (const [Npp8u](#) aValues[3], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.*
- [NppStatus nppiSet\\_8u\\_C4CR](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*4 channel 8-bit unsigned image set affecting only single channel.*

- [NppStatus nppiSet\\_16u\\_C1R](#) ([Npp16u](#) nValue, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*16-bit unsigned image set.*
- [NppStatus nppiSet\\_16u\\_C1MR](#) ([Npp16u](#) nValue, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 16-bit unsigned image set.*
- [NppStatus nppiSet\\_16u\\_C2R](#) (const [Npp16u](#) aValues[2], [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*2 channel 16-bit unsigned image set.*
- [NppStatus nppiSet\\_16u\\_C4R](#) (const [Npp16u](#) aValues[4], [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 16-bit unsigned image set.*
- [NppStatus nppiSet\\_16u\\_C4MR](#) (const [Npp16u](#) aValues[4], [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit unsigned image set.*
- [NppStatus nppiSet\\_16u\\_AC4R](#) (const [Npp16u](#) aValues[3], [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 16-bit unsigned image set method, not affecting Alpha channel.*
- [NppStatus nppiSet\\_16u\\_AC4MR](#) (const [Npp16u](#) aValues[3], [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.*
- [NppStatus nppiSet\\_16u\\_C4CR](#) ([Npp16u](#) nValue, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 16-bit unsigned image set affecting only single channel.*
- [NppStatus nppiSet\\_16s\\_C1R](#) ([Npp16s](#) nValue, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*16-bit image set.*
- [NppStatus nppiSet\\_16s\\_C1MR](#) ([Npp16s](#) nValue, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 16-bit image set.*
- [NppStatus nppiSet\\_16s\\_C2R](#) (const [Npp16s](#) aValues[2], [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*2 channel 16-bit image set.*
- [NppStatus nppiSet\\_16s\\_C4R](#) (const [Npp16s](#) aValues[4], [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 16-bit image set.*
- [NppStatus nppiSet\\_16s\\_C4MR](#) (const [Npp16s](#) aValues[4], [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set.*

- `NppStatus nppiSet_16s_AC4R` (const `Npp16s` aValues[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_16s_AC4MR` (const `Npp16s` aValues[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_16s_C4CR` (`Npp16s` nValue, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image set affecting only single channel.*
- `NppStatus nppiSet_16sc_C1R` (`Npp16sc` oValue, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer image set.*
- `NppStatus nppiSet_16sc_C2R` (`Npp16sc` aValue[2], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer two-channel image set.*
- `NppStatus nppiSet_16sc_C3R` (`Npp16sc` aValue[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer three-channel image set.*
- `NppStatus nppiSet_16sc_AC4R` (`Npp16sc` aValue[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer four-channel image set ignoring alpha.*
- `NppStatus nppiSet_16sc_C4R` (`Npp16sc` aValue[4], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex integer four-channel image set.*
- `NppStatus nppiSet_32s_C1R` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit image set.*
- `NppStatus nppiSet_32s_C1MR` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 32-bit image set.*
- `NppStatus nppiSet_32s_C4R` (const `Npp32s` aValues[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image set.*
- `NppStatus nppiSet_32s_C4MR` (const `Npp32s` aValues[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit image set.*
- `NppStatus nppiSet_32s_AC4R` (const `Npp32s` aValues[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image set method, not affecting Alpha channel.*

- `NppStatus nppiSet_32s_AC4MR` (const `Npp32s` aValues[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_32s_C4CR` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit unsigned image set affecting only single channel.*
- `NppStatus nppiSet_32sc_C1R` (`Npp32sc` oValue, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit complex integer image set.*
- `NppStatus nppiSet_32sc_C2R` (`Npp32sc` aValue[2], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two channel 32-bit complex integer image set.*
- `NppStatus nppiSet_32sc_C3R` (`Npp32sc` aValue[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit complex integer image set.*
- `NppStatus nppiSet_32sc_C4R` (`Npp32sc` aValue[4], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit complex integer image set.*
- `NppStatus nppiSet_32sc_AC4R` (`Npp32sc` aValue[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit complex integer four-channel image set ignoring alpha.*
- `NppStatus nppiSet_32f_C1R` (`Npp32f` nValue, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating point image set.*
- `NppStatus nppiSet_32f_C1MR` (`Npp32f` nValue, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 32-bit floating point image set.*
- `NppStatus nppiSet_32f_C4R` (const `Npp32f` aValues[4], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image set.*
- `NppStatus nppiSet_32f_C4MR` (const `Npp32f` aValues[4], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set.*
- `NppStatus nppiSet_32f_AC4R` (const `Npp32f` aValues[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_32f_AC4MR` (const `Npp32f` aValues[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_32f_C4CR` (`Npp32f` nValue, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

4 channel 32-bit floating point image set affecting only single channel.

- **NppStatus nppiSet\_32fc\_C1R** (**Npp32fc** oValue, **Npp32fc** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

Single channel 32-bit complex image set.

- **NppStatus nppiSet\_32fc\_C2R** (**Npp32fc** aValue[2], **Npp32fc** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

Two channel 32-bit complex image set.

- **NppStatus nppiSet\_32fc\_C3R** (**Npp32fc** aValue[3], **Npp32fc** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

Three channel 32-bit complex image set.

- **NppStatus nppiSet\_32fc\_C4R** (**Npp32fc** aValue[4], **Npp32fc** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

Four channel 32-bit complex image set.

- **NppStatus nppiSet\_32fc\_AC4R** (**Npp32fc** aValue[3], **Npp32fc** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

32-bit complex four-channel image set ignoring alpha.

## 7.54.1 Function Documentation

### 7.54.1.1 **NppStatus nppiSet\_16s\_AC4MR** (**const Npp16s** aValues[3], **Npp16s** \*pDst, **int** nDstStep, **NppiSize** oSizeROI, **const Npp8u** \*pMask, **int** nMaskStep)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

#### Parameters:

**aValues** Three-channel array containing the pixel-value to be set.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

**pMask** Pointer to the mask image. This is a single channel 8-bit unsigned int image.

**nMaskStep** Number of bytes between line starts of successive lines in the mask image.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.54.1.2 **NppStatus nppiSet\_16s\_AC4R** (**const Npp16s** aValues[3], **Npp16s** \*pDst, **int** nDstStep, **NppiSize** oSizeROI)

4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.3 NppStatus nppiSet\_16s\_C1MR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 16-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.4 NppStatus nppiSet\_16s\_C1R (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.5 NppStatus nppiSet\_16s\_C2R (const Npp16s aValues[2], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.6 NppStatus nppiSet\_16s\_C4CR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the pDst pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass pDst unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass pDst + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.7 NppStatus nppiSet\_16s\_C4MR (const Npp16s aValues[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 4 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.8 NppStatus nppiSet\_16s\_C4R (const Npp16s aValues[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.9 NppStatus nppiSet\_16sc\_AC4R (Npp16sc aValue[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.10 NppStatus nppiSet\_16sc\_C1R (Npp16sc oValue, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer image set.

**Parameters:**

*oValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.11 NppStatus nppiSet\_16sc\_C2R (Npp16sc aValue[2], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer two-channel image set.

**Parameters:**

- aValue* New pixel value.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.12 NppStatus nppiSet\_16sc\_C3R (Npp16sc aValue[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer three-channel image set.

**Parameters:**

- aValue* New pixel value.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.13 NppStatus nppiSet\_16sc\_C4R (Npp16sc aValue[4], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer four-channel image set.

**Parameters:**

- aValue* New pixel value.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.14** `NppStatus nppiSet_16u_AC4MR (const Npp16u aValues[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.15** `NppStatus nppiSet_16u_AC4R (const Npp16u aValues[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.16** `NppStatus nppiSet_16u_C1MR (Npp16u nValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 16-bit unsigned image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.17 NppStatus nppiSet\_16u\_C1R (Npp16u nValue, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit unsigned image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.18 NppStatus nppiSet\_16u\_C2R (const Npp16u aValues[2], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 16-bit unsigned image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.19 NppStatus nppiSet\_16u\_C4CR (Npp16u nValue, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the pDst pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass pDst unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass pDst + 2 to the function.

**Parameters:**

- nValue* The pixel-value to be set.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.1.20 NppStatus nppiSet\_16u\_C4MR (const Npp16u *aValues*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 4 channel 16-bit unsigned image set.

**Parameters:**

- aValues* New pixel value.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.
- nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.1.21 NppStatus nppiSet\_16u\_C4R (const Npp16u *aValues*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit unsigned image set.

**Parameters:**

- aValues* New pixel value.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.22** `NppStatus nppiSet_32f_AC4MR (const Npp32f aValues[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.23** `NppStatus nppiSet_32f_AC4R (const Npp32f aValues[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.24** `NppStatus nppiSet_32f_C1MR (Npp32f nValue, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit floating point image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.  
*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.1.25 NppStatus nppiSet\_32f\_C1R (Npp32f *nValue*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

32-bit floating point image set.

**Parameters:**

*nValue* New pixel value.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.1.26 NppStatus nppiSet\_32f\_C4CR (Npp32f *nValue*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit floating point image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.1.27 NppStatus nppiSet\_32f\_C4MR (const Npp32f *aValues*[4], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 4 channel 32-bit floating point image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.28 NppStatus nppiSet\_32f\_C4R (const Npp32f aValues[4], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit floating point image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.29 NppStatus nppiSet\_32fc\_AC4R (Npp32fc aValue[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit complex four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.30 NppStatus nppiSet\_32fc\_C1R (Npp32fc *oValue*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit complex image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.31 NppStatus nppiSet\_32fc\_C2R (Npp32fc *aValue*[2], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.32 NppStatus nppiSet\_32fc\_C3R (Npp32fc *aValue*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.33 NppStatus nppiSet\_32fc\_C4R (Npp32fc aValue[4], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 32-bit complex image set.

**Parameters:**

- aValue* The pixel-value to be set.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.34 NppStatus nppiSet\_32s\_AC4MR (const Npp32s aValues[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

- aValues* Three-channel array containing the pixel-value to be set.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.
- nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.35 NppStatus nppiSet\_32s\_AC4R (const Npp32s aValues[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

- aValues* Three-channel array containing the pixel-value to be set.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.36** `NppStatus nppiSet_32s_C1MR (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.37** `NppStatus nppiSet_32s_C1R (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit image set.

**Parameters:**

*nValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.38** `NppStatus nppiSet_32s_C4CR (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the pDst pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass pDst unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass pDst + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.39** `NppStatus nppiSet_32s_C4MR (const Npp32s aValues[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.40** `NppStatus nppiSet_32s_C4R (const Npp32s aValues[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit image set.

**Parameters:**

*aValues* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.41** `NppStatus nppiSet_32sc_AC4R (Npp32sc aValue[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* New pixel value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.42** `NppStatus nppiSet_32sc_C1R (Npp32sc oValue, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit complex integer image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.43** `NppStatus nppiSet_32sc_C2R (Npp32sc aValue[2], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Two channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.44** `NppStatus nppiSet_32sc_C3R (Npp32sc aValue[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.45** `NppStatus nppiSet_32sc_C4R (Npp32sc aValue[4], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.46** `NppStatus nppiSet_8s_AC4R (Npp8s aValue[3], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit four-channel image set ignoring alpha channel.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.47** `NppStatus nppiSet_8s_C1R (Npp8s nValue, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit image set.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.1.48 NppStatus nppiSet\_8s\_C2R (Npp8s *aValue*[2], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit two-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.49 NppStatus nppiSet\_8s\_C3R (Npp8s *aValue*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit three-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.50 NppStatus nppiSet\_8s\_C4R (Npp8s *aValue*[4], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit four-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.51 NppStatus nppiSet\_8u\_AC4MR (const Npp8u aValues[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.52 NppStatus nppiSet\_8u\_AC4R (const Npp8u aValues[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned image set method, not affecting Alpha channel.

For RGBA images, this method allows setting of the RGB values without changing the contents of the alpha-channel (fourth channel).

**Parameters:**

*aValues* Three-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.53 NppStatus nppiSet\_8u\_C1MR (Npp8u nValue, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 8-bit unsigned image set.

The 8-bit mask image affects setting of the respective pixels in the destination image. If the mask value is zero (0) the pixel is not set, if the mask is non-zero, the corresponding destination pixel is set to specified value.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* Pointer [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.54** `NppStatus nppiSet_8u_C1R (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit unsigned image set.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.55** `NppStatus nppiSet_8u_C4CR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set affecting only single channel.

For RGBA images, this method allows setting of a single of the four (RGBA) values without changing the contents of the other three channels. The channel is selected via the *pDst* pointer. The pointer needs to point to the actual first value to be set, e.g. in order to set the R-channel (first channel), one would pass *pDst* unmodified, since its value actually points to the r channel. If one wanted to modify the B channel (second channel), one would pass *pDst* + 2 to the function.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.56** `NppStatus nppiSet_8u_C4MR (const Npp8u aValues[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set.

**Parameters:**

*aValues* Four-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the mask image. This is a single channel 8-bit unsigned int image.

*nMaskStep* Number of bytes between line starts of successive lines in the mask image.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.1.57** `NppStatus nppiSet_8u_C4R (const Npp8u aValues[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set.

**Parameters:**

*aValues* Four-channel array containing the pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.55 Copy

### Image-Memory Copy

Copy methods for images of various types.

In addition to routines for copying pixels of identical layout from one image to another, there are copy routines for select channels as well as packed-planar conversions:

- Select channel to multi-channel copy. E.g. given a three-channel source and destination image one may copy the second channel of the source to the third channel of the destination.
- Single channel to multi-channel copy. E.g. given a single-channel source and a four-channel destination, one may copy the contents of the single-channel source to the second channel of the destination.
- Select channel to single-channel copy. E.g. given a three-channel source and a single-channel destination one may copy the third channel of the source to the destination.
- Multi-channel to planar copy. These copy operations split a multi-channel image into a set of single-channel images.
- Planar image to multi-channel copy. These copy routines combine separate color-planes (single channel images) into a single multi-channel image.
- [NppStatus nppiCopy\\_8s\\_C1R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit image copy.*
- [NppStatus nppiCopy\\_8s\\_C2R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Two-channel 8-bit image copy.*
- [NppStatus nppiCopy\\_8s\\_C3R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 8-bit image copy.*
- [NppStatus nppiCopy\\_8s\\_C4R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 8-bit image copy.*
- [NppStatus nppiCopy\\_8s\\_AC4R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 8-bit image copy, ignoring alpha channel.*
- [NppStatus nppiCopy\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*8-bit unsigned image copy.*
- [NppStatus nppiCopy\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*4 channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned image copy, not affecting Alpha channel.*
- `NppStatus nppiCopy_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit unsigned image copy.*
- `NppStatus nppiCopy_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image copy.*
- `NppStatus nppiCopy_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit unsigned image copy, not affecting Alpha channel.*
- `NppStatus nppiCopy_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit image copy.*
- `NppStatus nppiCopy_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image copy.*
- `NppStatus nppiCopy_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 16-bit image copy, not affecting Alpha.*
- `NppStatus nppiCopy_16sc_C1R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*16-bit complex image copy.*
- `NppStatus nppiCopy_16sc_C2R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 16-bit complex image copy.*
- `NppStatus nppiCopy_16sc_C3R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 16-bit complex image copy.*
- `NppStatus nppiCopy_16sc_C4R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 16-bit complex image copy.*
- `NppStatus nppiCopy_16sc_AC4R` (const `Npp16sc` \*pSrc, int nSrcStep, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 16-bit complex image copy, ignoring alpha.*
- `NppStatus nppiCopy_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit image copy.*

- `NppStatus nppiCopy_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image copy.*
- `NppStatus nppiCopy_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image copy, not affecting Alpha.*
- `NppStatus nppiCopy_32sc_C1R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C2R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C3R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_AC4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit complex image copy, ignoring alpha.*
- `NppStatus nppiCopy_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy, not affecting Alpha.*
- `NppStatus nppiCopy_32fc_C1R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C2R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C3R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit floating-point complex image copy.*

- `NppStatus nppiCopy_32fc_C4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit floating-point complex image copy.*

- `NppStatus nppiCopy_32fc_AC4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit floating-point complex image copy, ignoring alpha.*

- `NppStatus nppiCopy_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C3MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation three channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_AC4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.*

- `NppStatus nppiCopy_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C3MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation three channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_AC4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 16-bit unsigned image copy, ignoring alpha.*

- `NppStatus nppiCopy_16s_C1MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation 16-bit signed image copy.*

- `NppStatus nppiCopy_16s_C3MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation three channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16s_C4MR` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 16-bit signed image copy.*
- `NppStatus nppiCopy_16s_AC4MR` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 16-bit signed image copy, ignoring alpha.*
- `NppStatus nppiCopy_32s_C1MR` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation 32-bit signed image copy.*
- `NppStatus nppiCopy_32s_C3MR` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation three channel 32-bit signed image copy.*
- `NppStatus nppiCopy_32s_C4MR` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 32-bit signed image copy.*
- `NppStatus nppiCopy_32s_AC4MR` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 32-bit signed image copy, ignoring alpha.*
- `NppStatus nppiCopy_32f_C1MR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation 32-bit float image copy.*
- `NppStatus nppiCopy_32f_C3MR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation three channel 32-bit float image copy.*
- `NppStatus nppiCopy_32f_C4MR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 32-bit float image copy.*
- `NppStatus nppiCopy_32f_AC4MR` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u *pMask`, int `nMaskStep`)  
*Masked Operation four channel 32-bit float image copy, ignoring alpha.*
- `NppStatus nppiCopy_8u_C3CR` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Select-channel 8-bit unsigned image copy for three-channel images.*
- `NppStatus nppiCopy_8u_C4CR` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Select-channel 8-bit unsigned image copy for four-channel images.*
- `NppStatus nppiCopy_16s_C3CR` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*Select-channel 16-bit signed image copy for three-channel images.*

- [NppStatus nppiCopy\\_16s\\_C4CR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 16-bit signed image copy for four-channel images.*
- [NppStatus nppiCopy\\_16u\\_C3CR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 16-bit unsigned image copy for three-channel images.*
- [NppStatus nppiCopy\\_16u\\_C4CR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 16-bit unsigned image copy for four-channel images.*
- [NppStatus nppiCopy\\_32s\\_C3CR](#) (const [Npp32s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 32-bit signed image copy for three-channel images.*
- [NppStatus nppiCopy\\_32s\\_C4CR](#) (const [Npp32s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 32-bit signed image copy for four-channel images.*
- [NppStatus nppiCopy\\_32f\\_C3CR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 32-bit float image copy for three-channel images.*
- [NppStatus nppiCopy\\_32f\\_C4CR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Select-channel 32-bit float image copy for four-channel images.*
- [NppStatus nppiCopy\\_8u\\_C3C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three-channel to single-channel 8-bit unsigned image copy.*
- [NppStatus nppiCopy\\_8u\\_C4C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four-channel to single-channel 8-bit unsigned image copy.*
- [NppStatus nppiCopy\\_16s\\_C3C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three-channel to single-channel 16-bit signed image copy.*
- [NppStatus nppiCopy\\_16s\\_C4C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four-channel to single-channel 16-bit signed image copy.*
- [NppStatus nppiCopy\\_16u\\_C3C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three-channel to single-channel 16-bit unsigned image copy.*
- [NppStatus nppiCopy\\_16u\\_C4C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four-channel to single-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_32s_C3C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32s_C4C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32f_C3C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 32-bit float image copy.*

- `NppStatus nppiCopy_32f_C4C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 32-bit float image copy.*

- `NppStatus nppiCopy_8u_C1C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C1C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_16s_C1C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16s_C1C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16u_C1C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C1C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_32s_C1C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32s_C1C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 32-bit signed image copy.*

- **NppStatus nppiCopy\_32f\_C1C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to three-channel 32-bit float image copy.*
- **NppStatus nppiCopy\_32f\_C1C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single-channel to four-channel 32-bit float image copy.*
- **NppStatus nppiCopy\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_8u\_C4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_16s\_C3P3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_16s\_C4P4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_16u\_C3P3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_16u\_C4P4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned packed to planar image copy.*
- **NppStatus nppiCopy\_32s\_C3P3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_32s\_C4P4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit signed packed to planar image copy.*
- **NppStatus nppiCopy\_32f\_C3P3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*const aDst[3], int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit float packed to planar image copy.*
- **NppStatus nppiCopy\_32f\_C4P4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*const aDst[4], int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit float packed to planar image copy.*
- **NppStatus nppiCopy\_8u\_P3C3R** (const **Npp8u** \*const aSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned planar to packed image copy.*

- `NppStatus nppiCopy_8u_P4C4R` (const `Npp8u` \*const `aSrc[4]`, int `nSrcStep`, `Npp8u` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four-channel 8-bit unsigned planar to packed image copy.*
- `NppStatus nppiCopy_16u_P3C3R` (const `Npp16u` \*const `aSrc[3]`, int `nSrcStep`, `Npp16u` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three-channel 16-bit unsigned planar to packed image copy.*
- `NppStatus nppiCopy_16u_P4C4R` (const `Npp16u` \*const `aSrc[4]`, int `nSrcStep`, `Npp16u` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four-channel 16-bit unsigned planar to packed image copy.*
- `NppStatus nppiCopy_16s_P3C3R` (const `Npp16s` \*const `aSrc[3]`, int `nSrcStep`, `Npp16s` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three-channel 16-bit signed planar to packed image copy.*
- `NppStatus nppiCopy_16s_P4C4R` (const `Npp16s` \*const `aSrc[4]`, int `nSrcStep`, `Npp16s` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four-channel 16-bit signed planar to packed image copy.*
- `NppStatus nppiCopy_32s_P3C3R` (const `Npp32s` \*const `aSrc[3]`, int `nSrcStep`, `Npp32s` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three-channel 32-bit signed planar to packed image copy.*
- `NppStatus nppiCopy_32s_P4C4R` (const `Npp32s` \*const `aSrc[4]`, int `nSrcStep`, `Npp32s` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four-channel 32-bit signed planar to packed image copy.*
- `NppStatus nppiCopy_32f_P3C3R` (const `Npp32f` \*const `aSrc[3]`, int `nSrcStep`, `Npp32f` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three-channel 32-bit float planar to packed image copy.*
- `NppStatus nppiCopy_32f_P4C4R` (const `Npp32f` \*const `aSrc[4]`, int `nSrcStep`, `Npp32f` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four-channel 32-bit float planar to packed image copy.*

## 7.55.1 Function Documentation

### 7.55.1.1 `NppStatus nppiCopy_16s_AC4MR` (const `Npp16s` \*`pSrc`, int `nSrcStep`, `Npp16s` \*`pDst`, int `nDstStep`, `NppiSize` `oSizeROI`, const `Npp8u` \*`pMask`, int `nMaskStep`)

**Masked Operation** four channel 16-bit signed image copy, ignoring alpha.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.2 NppStatus nppiCopy\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.3 NppStatus nppiCopy\_16s\_C1C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to three-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.55.1.4 NppStatus nppiCopy\_16s\_C1C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)

Single-channel to four-channel 16-bit signed image copy.

##### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.55.1.5 NppStatus nppiCopy\_16s\_C1MR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)

Masked Operation 16-bit signed image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.55.1.6 NppStatus nppiCopy\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)

16-bit image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.7 NppStatus nppiCopy\_16s\_C3C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel to single-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.8 NppStatus nppiCopy\_16s\_C3CR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 16-bit signed image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.9 NppStatus nppiCopy\_16s\_C3MR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation three channel 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.10** `NppStatus nppiCopy_16s_C3P3R (const Npp16s * pSrc, int nSrcStep, Npp16s * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.11** `NppStatus nppiCopy_16s_C4C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.12** `NppStatus nppiCopy_16s_C4CR (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 16-bit signed image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.13** `NppStatus nppiCopy_16s_C4MR (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.14** `NppStatus nppiCopy_16s_C4P4R (const Npp16s * pSrc, int nSrcStep, Npp16s *const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.15** `NppStatus nppiCopy_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.16** `NppStatus nppiCopy_16s_P3C3R (const Npp16s *const aSrc[3], int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.17** `NppStatus nppiCopy_16s_P4C4R (const Npp16s *const aSrc[4], int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.18** `NppStatus nppiCopy_16sc_AC4R (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.19 NppStatus nppiCopy\_16sc\_C1R (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.20 NppStatus nppiCopy\_16sc\_C2R (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.21 NppStatus nppiCopy\_16sc\_C3R (const Npp16sc \* pSrc, int nSrcStep, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.22** `NppStatus nppiCopy_16sc_C4R (const Npp16sc * pSrc, int nSrcStep, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.23** `NppStatus nppiCopy_16u_AC4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 16-bit unsigned image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.24** `NppStatus nppiCopy_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.25 NppStatus nppiCopy\_16u\_C1C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to three-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.26 NppStatus nppiCopy\_16u\_C1C4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.27 NppStatus nppiCopy\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.28 NppStatus nppiCopy\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit unsigned image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.29 NppStatus nppiCopy\_16u\_C3C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

- pSrc* Select-Channel Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.30 NppStatus nppiCopy\_16u\_C3CR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 16-bit unsigned image copy for three-channel images.

**Parameters:**

- pSrc* Select-Channel Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Select-Channel Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.31** `NppStatus nppiCopy_16u_C3MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.32** `NppStatus nppiCopy_16u_C3P3R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.33** `NppStatus nppiCopy_16u_C4C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.34** `NppStatus nppiCopy_16u_C4CR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 16-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.35** `NppStatus nppiCopy_16u_C4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.36** `NppStatus nppiCopy_16u_C4P4R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.37 NppStatus nppiCopy\_16u\_C4R (const Npp16u \*pSrc, int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.38 NppStatus nppiCopy\_16u\_P3C3R (const Npp16u \*const aSrc[3], int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 16-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.39 NppStatus nppiCopy\_16u\_P4C4R (const Npp16u \*const aSrc[4], int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 16-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.40** `NppStatus nppiCopy_32f_AC4MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 32-bit float image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.41** `NppStatus nppiCopy_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.42** `NppStatus nppiCopy_32f_C1C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to three-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.43 NppStatus nppiCopy\_32f\_C1C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.44 NppStatus nppiCopy\_32f\_C1MR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.45 NppStatus nppiCopy\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.46** `NppStatus nppiCopy_32f_C3C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.47** `NppStatus nppiCopy_32f_C3CR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 32-bit float image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.48** `NppStatus nppiCopy_32f_C3MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.49 NppStatus nppiCopy\_32f\_C3P3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \*const aDst[3], int nDstStep, NppiSize oSizeROI)**

Three-channel 32-bit float packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.50 NppStatus nppiCopy\_32f\_C4C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel to single-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.51 NppStatus nppiCopy\_32f\_C4CR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 32-bit float image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.52** `NppStatus nppiCopy_32f_C4MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.53** `NppStatus nppiCopy_32f_C4P4R (const Npp32f * pSrc, int nSrcStep, Npp32f *const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit float packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.54** `NppStatus nppiCopy_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.55** `NppStatus nppiCopy_32f_P3C3R (const Npp32f *const aSrc[3], int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 32-bit float planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.56** `NppStatus nppiCopy_32f_P4C4R (const Npp32f *const aSrc[4], int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit float planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.57** `NppStatus nppiCopy_32fc_AC4R (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.58 NppStatus nppiCopy\_32fc\_C1R (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit floating-point complex image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.59 NppStatus nppiCopy\_32fc\_C2R (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two-channel 32-bit floating-point complex image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.60 NppStatus nppiCopy\_32fc\_C3R (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 32-bit floating-point complex image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.61 NppStatus nppiCopy\_32fc\_C4R (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.62 NppStatus nppiCopy\_32s\_AC4MR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

[Masked Operation](#) four channel 32-bit signed image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.63 NppStatus nppiCopy\_32s\_AC4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.64 NppStatus nppiCopy\_32s\_C1C3R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to three-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.65 NppStatus nppiCopy\_32s\_C1C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.66 NppStatus nppiCopy\_32s\_C1MR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.67 NppStatus nppiCopy\_32s\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.68 NppStatus nppiCopy\_32s\_C3C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel to single-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.69 NppStatus nppiCopy\_32s\_C3CR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 32-bit signed image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.70** `NppStatus nppiCopy_32s_C3MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.71** `NppStatus nppiCopy_32s_C3P3R (const Npp32s * pSrc, int nSrcStep, Npp32s * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 32-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.72** `NppStatus nppiCopy_32s_C4C1R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.73** `NppStatus nppiCopy_32s_C4CR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 32-bit signed image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.74** `NppStatus nppiCopy_32s_C4MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.75** `NppStatus nppiCopy_32s_C4P4R (const Npp32s * pSrc, int nSrcStep, Npp32s * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.76 NppStatus nppiCopy\_32s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.77 NppStatus nppiCopy\_32s\_P3C3R (const Npp32s \*const aSrc[3], int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.78 NppStatus nppiCopy\_32s\_P4C4R (const Npp32s \*const aSrc[4], int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.79 NppStatus nppiCopy\_32sc\_AC4R (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.80 NppStatus nppiCopy\_32sc\_C1R (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.81 NppStatus nppiCopy\_32sc\_C2R (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.82** `NppStatus nppiCopy_32sc_C3R (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 32-bit complex image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.83** `NppStatus nppiCopy_32sc_C4R (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit complex image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.84** `NppStatus nppiCopy_8s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit image copy, ignoring alpha channel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.85** `NppStatus nppiCopy_8s_C1R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.86** `NppStatus nppiCopy_8s_C2R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Two-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.87** `NppStatus nppiCopy_8s_C3R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.88 NppStatus nppiCopy\_8s\_C4R (const Npp8s \* pSrc, int nSrcStep, Npp8s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.89 NppStatus nppiCopy\_8u\_AC4MR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

[Masked Operation](#) four channel 8-bit unsigned image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.90 NppStatus nppiCopy\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.91** `NppStatus nppiCopy_8u_C1C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to three-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.92** `NppStatus nppiCopy_8u_C1C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single-channel to four-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.93** `NppStatus nppiCopy_8u_C1MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.94 NppStatus nppiCopy\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

8-bit unsigned image copy.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.95 NppStatus nppiCopy\_8u\_C3C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

- pSrc* Select-Channel Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.96 NppStatus nppiCopy\_8u\_C3CR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 8-bit unsigned image copy for three-channel images.

**Parameters:**

- pSrc* Select-Channel Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Select-Channel Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.97** `NppStatus nppiCopy_8u_C3MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.98** `NppStatus nppiCopy_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.99** `NppStatus nppiCopy_8u_C4C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.100 NppStatus nppiCopy\_8u\_C4CR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 8-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.101 NppStatus nppiCopy\_8u\_C4MR (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation four channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.102 NppStatus nppiCopy\_8u\_C4P4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* const aDst[4], int nDstStep, NppiSize oSizeROI)**

Four-channel 8-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Planar Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.103 NppStatus nppiCopy\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.104 NppStatus nppiCopy\_8u\_P3C3R (const Npp8u \*const aSrc[3], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 8-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.105 NppStatus nppiCopy\_8u\_P4C4R (const Npp8u \*const aSrc[4], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 8-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.56 Convert

### Bit-Depth Conversion

Convert bit-depth up and down.

The integer conversion methods do not involve any scaling. Conversions that reduce bit-depth saturate values exceeding the reduced range to the range's maximum/minimum value. When converting from floating-point values to integer values, a rounding mode can be specified. After rounding to integer values the values get saturated to the destination data type's range.

- `NppStatus nppiConvert_8u16u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiConvert_8u16u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiConvert_8u16u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiConvert_8u16u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_8u16s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiConvert_8u16s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiConvert_8u16s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiConvert_8u16s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.*
- `NppStatus nppiConvert_8u32s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned to 32-bit signed conversion.*
- `NppStatus nppiConvert_8u32s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned to 32-bit signed conversion.*

- **NppStatus nppiConvert\_8u32s\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 32-bit signed conversion.*
- **NppStatus nppiConvert\_8u32s\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8u32f\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 32-bit floating-point conversion.*
- **NppStatus nppiConvert\_8u32f\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit unsigned to 32-bit floating-point conversion.*
- **NppStatus nppiConvert\_8u32f\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 32-bit floating-point conversion.*
- **NppStatus nppiConvert\_8u32f\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8s32s\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit signed to 32-bit signed conversion.*
- **NppStatus nppiConvert\_8s32s\_C3R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit signed to 32-bit signed conversion.*
- **NppStatus nppiConvert\_8s32s\_C4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit signed to 32-bit signed conversion.*
- **NppStatus nppiConvert\_8s32s\_AC4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8s32f\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit signed to 32-bit floating-point conversion.*
- **NppStatus nppiConvert\_8s32f\_C3R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit signed to 32-bit floating-point conversion.*
- **NppStatus nppiConvert\_8s32f\_C4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit signed to 32-bit floating-point conversion.*

- [NppStatus nppiConvert\\_8s32f\\_AC4R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_16u32s\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Single channel 16-bit unsigned to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16u32s\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three channel 16-bit unsigned to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16u32s\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 16-bit unsigned to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16u32s\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_16u32f\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Single channel 16-bit unsigned to 32-bit floating-point conversion.*
- [NppStatus nppiConvert\\_16u32f\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three channel 16-bit unsigned to 32-bit floating-point conversion.*
- [NppStatus nppiConvert\\_16u32f\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 16-bit unsigned to 32-bit floating-point conversion.*
- [NppStatus nppiConvert\\_16u32f\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_16s32s\\_C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Single channel 16-bit signed to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16s32s\\_C3R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Three channel 16-bit signed to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16s32s\\_C4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)  
*Four channel 16-bit signed to 32-bit signed conversion.*
- [NppStatus nppiConvert\\_16s32s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.*

- `NppStatus nppiConvert_16s32f_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_16s32f_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_16s32f_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_16s32f_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.*

- `NppStatus nppiConvert_8s8u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 8-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_8s16u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 16-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_8s16s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 16-bit signed conversion.*

- `NppStatus nppiConvert_8s32u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 32-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_16s16u_C1Rs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed to 16-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_16s32u_C1Rs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed to 32-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_16u32u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit unsigned to 32-bit unsigned conversion.*

- `NppStatus nppiConvert_32s32u_C1Rs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit signed to 32-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_32s32f_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit signed to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_32u32f_C1R` (const `Npp32u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit unsigned to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16u8u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit unsigned to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16u8u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit unsigned to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16u8u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16u8u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16s8u_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16s8u_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit signed to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16s8u_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_16s8u_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_32s8u_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit signed to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_32s8u_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit signed to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_32s8u_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_32s8u_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiConvert_32s8s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit signed to 8-bit signed conversion.*
- `NppStatus nppiConvert_32s8s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit signed to 8-bit signed conversion.*
- `NppStatus nppiConvert_32s8s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit signed to 8-bit signed conversion.*
- `NppStatus nppiConvert_32s8s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.*
- `NppStatus nppiConvert_8u8s_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_16u8s_C1RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_16s8s_C1RSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_16u16s_C1RSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32u8u_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32u8s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32u16u_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32u16s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32u32s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32s16u_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32s16s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppiConvert_32f8u_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)  
*Single channel 32-bit floating point to 8-bit unsigned conversion.*
- `NppStatus nppiConvert_32f8u_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)  
*Three channel 32-bit floating point to 8-bit unsigned conversion.*

- [NppStatus nppiConvert\\_32f8u\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 8-bit unsigned conversion.*
- [NppStatus nppiConvert\\_32f8u\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_32f8s\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Single channel 32-bit floating point to 8-bit signed conversion.*
- [NppStatus nppiConvert\\_32f8s\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Three channel 32-bit floating point to 8-bit signed conversion.*
- [NppStatus nppiConvert\\_32f8s\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 8-bit signed conversion.*
- [NppStatus nppiConvert\\_32f8s\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp8s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_32f16u\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Single channel 32-bit floating point to 16-bit unsigned conversion.*
- [NppStatus nppiConvert\\_32f16u\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Three channel 32-bit floating point to 16-bit unsigned conversion.*
- [NppStatus nppiConvert\\_32f16u\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 16-bit unsigned conversion.*
- [NppStatus nppiConvert\\_32f16u\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.*
- [NppStatus nppiConvert\\_32f16s\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Single channel 32-bit floating point to 16-bit signed conversion.*
- [NppStatus nppiConvert\\_32f16s\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Three channel 32-bit floating point to 16-bit signed conversion.*
- [NppStatus nppiConvert\\_32f16s\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppRoundMode](#) eRoundMode)  
*Four channel 32-bit floating point to 16-bit signed conversion.*

- **NppStatus nppiConvert\_32f16s\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus nppiConvert\_32f8u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f8s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 8-bit signed conversion.*
- **NppStatus nppiConvert\_32f16u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f16s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus nppiConvert\_32f32u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 32-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f32s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 32-bit signed conversion.*

## 7.56.1 Function Documentation

### 7.56.1.1 **NppStatus nppiConvert\_16s16u\_C1Rs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Single channel 16-bit signed to 16-bit unsigned conversion with saturation.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.56.1.2 NppStatus nppiConvert\_16s32f\_AC4R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)

Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.56.1.3 NppStatus nppiConvert\_16s32f\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)

Single channel 16-bit signed to 32-bit floating-point conversion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.56.1.4 NppStatus nppiConvert\_16s32f\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)

Three channel 16-bit signed to 32-bit floating-point conversion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.5 NppStatus nppiConvert\_16s32f\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.6 NppStatus nppiConvert\_16s32s\_AC4R (const Npp16s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.7 NppStatus nppiConvert\_16s32s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.8 NppStatus nppiConvert\_16s32s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.9 NppStatus nppiConvert\_16s32s\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.10 NppStatus nppiConvert\_16s32u\_C1Rs (const Npp16s \* pSrc, int nSrcStep, Npp32u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.11** `NppStatus nppiConvert_16s8s_C1RSfs (const Npp16s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

**7.56.1.12** `NppStatus nppiConvert_16s8u_AC4R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.13** `NppStatus nppiConvert_16s8u_C1R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.14** `NppStatus nppiConvert_16s8u_C3R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 16-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.15** `NppStatus nppiConvert_16s8u_C4R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.16** `NppStatus nppiConvert_16u16s_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

**7.56.1.17** `NppStatus nppiConvert_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.18** `NppStatus nppiConvert_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.19 NppStatus nppiConvert\_16u32f\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.20 NppStatus nppiConvert\_16u32f\_C4R (const Npp16u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.21 NppStatus nppiConvert\_16u32s\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.22 NppStatus nppiConvert\_16u32s\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.23 NppStatus nppiConvert\_16u32s\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.24** `NppStatus nppiConvert_16u32s_C4R (const Npp16u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.25** `NppStatus nppiConvert_16u32u_C1R (const Npp16u * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 32-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.26** `NppStatus nppiConvert_16u8s_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`**7.56.1.27** `NppStatus nppiConvert_16u8u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.28** `NppStatus nppiConvert_16u8u_C1R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.29** `NppStatus nppiConvert_16u8u_C3R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.30** `NppStatus nppiConvert_16u8u_C4R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.31 NppStatus nppiConvert\_32f16s\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Four channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.32 NppStatus nppiConvert\_32f16s\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Single channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.33** `NppStatus nppiConvert_32f16s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.34** `NppStatus nppiConvert_32f16s_C3R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.35** `NppStatus nppiConvert_32f16s_C4R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.36** `NppStatus nppiConvert_32f16u_AC4R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.37** `NppStatus nppiConvert_32f16u_C1R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.38** `NppStatus nppiConvert_32f16u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.39** `NppStatus nppiConvert_32f16u_C3R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.40** `NppStatus nppiConvert_32f16u_C4R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.41** `NppStatus nppiConvert_32f32s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.42** `NppStatus nppiConvert_32f32u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 32-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.43** `NppStatus nppiConvert_32f8s_AC4R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.44** `NppStatus nppiConvert_32f8s_C1R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.45** `NppStatus nppiConvert_32f8s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.46** `NppStatus nppiConvert_32f8s_C3R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.47** `NppStatus nppiConvert_32f8s_C4R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.48** `NppStatus nppiConvert_32f8u_AC4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.49** `NppStatus nppiConvert_32f8u_C1R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.50** `NppStatus nppiConvert_32f8u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.51 NppStatus nppiConvert\_32f8u\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Three channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.52 NppStatus nppiConvert\_32f8u\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Four channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- 7.56.1.53** `NppStatus nppiConvert_32s16s_C1RSfs` (const `Npp32s * pSrc`, int `nSrcStep`, `Npp16s * pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppRoundMode eRoundMode`, int `nScaleFactor`)
- 7.56.1.54** `NppStatus nppiConvert_32s16u_C1RSfs` (const `Npp32s * pSrc`, int `nSrcStep`, `Npp16u * pDst`, int `nDstStep`, `NppiSize oSizeROI`, `NppRoundMode eRoundMode`, int `nScaleFactor`)
- 7.56.1.55** `NppStatus nppiConvert_32s32f_C1R` (const `Npp32s * pSrc`, int `nSrcStep`, `Npp32f * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Single channel 32-bit signed to 32-bit floating-point conversion.

**Parameters:**

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.56.1.56** `NppStatus nppiConvert_32s32u_C1Rs` (const `Npp32s * pSrc`, int `nSrcStep`, `Npp32u * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Single channel 32-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.56.1.57** `NppStatus nppiConvert_32s8s_AC4R` (const `Npp32s * pSrc`, int `nSrcStep`, `Npp8s * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.

**Parameters:**

- `pSrc` Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.58** `NppStatus nppiConvert_32s8s_C1R (const Npp32s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.59** `NppStatus nppiConvert_32s8s_C3R (const Npp32s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.60 NppStatus nppiConvert\_32s8s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp8s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.61 NppStatus nppiConvert\_32s8u\_AC4R (const Npp32s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.62 NppStatus nppiConvert\_32s8u\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.63** `NppStatus nppiConvert_32s8u_C3R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.64** `NppStatus nppiConvert_32s8u_C4R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.65** `NppStatus nppiConvert_32u16s_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

**7.56.1.66** `NppStatus nppiConvert_32u16u_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

**7.56.1.67** `NppStatus nppiConvert_32u32f_C1R (const Npp32u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- 7.56.1.68** `NppStatus nppiConvert_32u32s_C1RSfs` (const `Npp32u * pSrc`, `int nSrcStep`, `Npp32s * pDst`, `int nDstStep`, `NppiSize oSizeROI`, `NppRoundMode eRoundMode`, `int nScaleFactor`)
- 7.56.1.69** `NppStatus nppiConvert_32u8s_C1RSfs` (const `Npp32u * pSrc`, `int nSrcStep`, `Npp8s * pDst`, `int nDstStep`, `NppiSize oSizeROI`, `NppRoundMode eRoundMode`, `int nScaleFactor`)
- 7.56.1.70** `NppStatus nppiConvert_32u8u_C1RSfs` (const `Npp32u * pSrc`, `int nSrcStep`, `Npp8u * pDst`, `int nDstStep`, `NppiSize oSizeROI`, `NppRoundMode eRoundMode`, `int nScaleFactor`)
- 7.56.1.71** `NppStatus nppiConvert_8s16s_C1R` (const `Npp8s * pSrc`, `int nSrcStep`, `Npp16s * pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Single channel 8-bit signed to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- 7.56.1.72** `NppStatus nppiConvert_8s16u_C1Rs` (const `Npp8s * pSrc`, `int nSrcStep`, `Npp16u * pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Single channel 8-bit signed to 16-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.73** `NppStatus nppiConvert_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.74** `NppStatus nppiConvert_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.75** `NppStatus nppiConvert_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.76 **NppStatus nppiConvert\_8s32f\_C4R** (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.77 **NppStatus nppiConvert\_8s32s\_AC4R** (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.78 **NppStatus nppiConvert\_8s32s\_C1R** (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.79** `NppStatus nppiConvert_8s32s_C3R (const Npp8s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.80** `NppStatus nppiConvert_8s32s_C4R (const Npp8s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.81 NppStatus nppiConvert\_8s32u\_C1Rs (const Npp8s \* pSrc, int nSrcStep, Npp32u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 8-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.82 NppStatus nppiConvert\_8s8u\_C1Rs (const Npp8s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 8-bit signed to 8-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.83 NppStatus nppiConvert\_8u16s\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.84** `NppStatus nppiConvert_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.85** `NppStatus nppiConvert_8u16s_C3R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.86** `NppStatus nppiConvert_8u16s_C4R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.87** `NppStatus nppiConvert_8u16u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.88** `NppStatus nppiConvert_8u16u_C1R (const Npp8u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.89** `NppStatus nppiConvert_8u16u_C3R (const Npp8u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.90 NppStatus nppiConvert\_8u16u\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.91 NppStatus nppiConvert\_8u32f\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.92 NppStatus nppiConvert\_8u32f\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.93 NppStatus nppiConvert\_8u32f\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.94 NppStatus nppiConvert\_8u32f\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.95 NppStatus nppiConvert\_8u32s\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.96 NppStatus nppiConvert\_8u32s\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.97 NppStatus nppiConvert\_8u32s\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.98 NppStatus nppiConvert\_8u32s\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.99** `NppStatus nppiConvert_8u8s_C1RSfs (const Npp8u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

## 7.57 Copy Constant Border

### Copy Const Border

Methods for copying images and padding borders with a constant, user-specifiable color.

- `NppStatus nppiCopyConstBorder_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSrcSizeROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oDstSizeROI`, int `nTopBorderHeight`, int `nLeftBorderWidth`, `Npp8u nValue`)  
*8-bit unsigned image copy width constant border color.*
- `NppStatus nppiCopyConstBorder_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSrcSizeROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oDstSizeROI`, int `nTopBorderHeight`, int `nLeftBorderWidth`, const `Npp8u aValue[4]`)  
*4channel 8-bit unsigned image copy with constant border color.*
- `NppStatus nppiCopyConstBorder_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSrcSizeROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oDstSizeROI`, int `nTopBorderHeight`, int `nLeftBorderWidth`, const `Npp8u aValue[3]`)  
*4 channel 8-bit unsigned image copy with constant border color.*
- `NppStatus nppiCopyConstBorder_32s_C1R` (const `Npp32s *pSrc`, int `nSrcStep`, `NppiSize oSrcSizeROI`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oDstSizeROI`, int `nTopBorderHeight`, int `nLeftBorderWidth`, `Npp32s nValue`)  
*32-bit image copy with constant border color.*

### 7.57.1 Function Documentation

**7.57.1.1** `NppStatus nppiCopyConstBorder_32s_C1R` (const `Npp32s *pSrc`, int `nSrcStep`, `NppiSize oSrcSizeROI`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oDstSizeROI`, int `nTopBorderHeight`, int `nLeftBorderWidth`, `Npp32s nValue`)

32-bit image copy with constant border color.

See `nppiCopyConstBorder_8u_C1R()` for detailed documentation.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `oSrcSizeROI` Size of the source region-of-interest.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oDstSizeROI` Size of the destination region-of-interest.
- `nTopBorderHeight` Height of top border.
- `nLeftBorderWidth` Width of left border.
- `nValue` Border luminance value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.2 NppStatus nppiCopyConstBorder\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])**

4 channel 8-bit unsigned image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.3 NppStatus nppiCopyConstBorder\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp8u nValue)**

8-bit unsigned image copy with constant border color.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region of pixels.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The height of the border at the bottom of the destination ROI is implicitly defined by the size of the source ROI:  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

*nValue* The pixel value to be set for border pixels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.4 NppStatus nppiCopyConstBorder\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[4])**

4channel 8-bit unsigned image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.58 Transpose And Swap Channels

### Image Transpose

Methods for transposing images of various types.

Like matrix transpose, image transpose is a mirror along the image's diagonal (upper-left to lower-right corner).

- `NppStatus nppiTranspose_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oROI`)

*8-bit image transpose.*

### Image Color Channel Swap

Methods for exchanging the color channels of an image.

The methods support arbitrary permutations of the original channels, including replication.

- `NppStatus nppiSwapChannels_8u_C4IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const int `aDstOrder[4]`)

*4 channel 8-bit unsigned swap channels, in-place.*

#### 7.58.1 Function Documentation

##### 7.58.1.1 `NppStatus nppiSwapChannels_8u_C4IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const int `aDstOrder[4]`)

4 channel 8-bit unsigned swap channels, in-place.

##### Parameters:

*`pSrcDst`* In-Place Image Pointer.

*`nSrcDstStep`* In-Place-Image Line Step.

*`oSizeROI`* Region-of-Interest (ROI).

*`aDstOrder`* Integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, `aDstOrder = [3,2,1,0]` converts this to ABGR channel order.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

##### 7.58.1.2 `NppStatus nppiTranspose_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oROI`)

8-bit image transpose.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Pointer to the destination ROI.
- nDstStep* Destination-Image Line Step.
- oROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.59 Filtering Functions

Linear and non-linear image filtering functions.

### Modules

- [1D Linear Filter](#)
- [1D Window Sum](#)
- [Convolution](#)
- [2D Fixed Linear Filters](#)
- [Rank Filters](#)

### 7.59.1 Detailed Description

Linear and non-linear image filtering functions.

## 7.60 1D Linear Filter

### 1D Linear Filter

1D mask Linear Convolution Filter, with rescaling, for 8 bit images.

- `NppStatus nppiFilterColumn_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s *pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

*8-bit unsigned 1D (column) image convolution.*

- `NppStatus nppiFilterColumn_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s *pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

*4 channel 8-bit unsigned 1D (column) image convolution.*

- `NppStatus nppiFilterRow_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s *pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

*8-bit unsigned 1D (row) image convolution.*

- `NppStatus nppiFilterRow_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s *pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

*4 channel 8-bit unsigned 1D (row) image convolution.*

### 7.60.1 Function Documentation

**7.60.1.1** `NppStatus nppiFilterColumn_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s *pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

8-bit unsigned 1D (column) image convolution.

Apply convolution filter with user specified 1D column of weights. Result pixel is equal to the sum of the products between the kernel coefficients (`pKernel` array) and corresponding neighboring column pixel values in the source image defined by `nKernelDim` and `nAnchorY`, divided by `nDivisor`.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.2 NppStatus nppiFilterColumn\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s \* pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)**

4 channel 8-bit unsigned 1D (column) image convolution.

Apply convolution filter with user specified 1D column of weights. Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.3 NppStatus nppiFilterRow\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s \* pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)**

8-bit unsigned 1D (row) image convolution.

Apply general linear Row convolution filter, with rescaling, in a 1D mask region around each source pixel for 1-channel 8 bit/pixel images. Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring row pixel values in the source image defined by iKernelDim and iAnchorX, divided by iDivisor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.4 NppStatus nppiFilterRow\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s \* pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)**

4 channel 8-bit unsigned 1D (row) image convolution.

Apply general linear Row convolution filter, with rescaling, in a 1D mask region around each source pixel for 1-channel 8 bit/pixel images. Result pixel is equal to the sum of the products between the kernel coefficients (*pKernel* array) and corresponding neighboring row pixel values in the source image defined by *iKernelDim* and *iAnchorX*, divided by *iDivisor*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.61 1D Window Sum

### 1D Window Sum

1D mask Window Sum for 8 bit images.

- `NppStatus nppiSumWindowColumn_8u32f_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, `Npp32s nMaskSize`, `Npp32s nAnchor`)  
8-bit unsigned 1D (column) sum to 32f.
- `NppStatus nppiSumWindowRow_8u32f_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, `Npp32s nMaskSize`, `Npp32s nAnchor`)  
8-bit unsigned 1D (row) sum to 32f.

#### 7.61.1 Function Documentation

##### 7.61.1.1 `NppStatus nppiSumWindowColumn_8u32f_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, `Npp32s nMaskSize`, `Npp32s nAnchor`)

8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by `nMaskSize` and `nAnchor`.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oROI` Region-of-Interest (ROI).
- `nMaskSize` Length of the linear kernel array.
- `nAnchor` Y offset of the kernel origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

##### 7.61.1.2 `NppStatus nppiSumWindowRow_8u32f_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oROI`, `Npp32s nMaskSize`, `Npp32s nAnchor`)

8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by `iKernelDim` and `iAnchorX`.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.62 Convolution

### Convolution (2D Masks)

General purpose 2D convolution filters.

- `NppStatus nppiFilter_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, const `Npp32s *pKernel`, `NppiSize oKernelSize`, `NppiPoint oAnchor`, `Npp32s nDivisor`)

*8-bit unsigned convolution filter.*

- `NppStatus nppiFilter_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, const `Npp32s *pKernel`, `NppiSize oKernelSize`, `NppiPoint oAnchor`, `Npp32s nDivisor`)

*4 channel 8-bit unsigned convolution filter.*

#### 7.62.1 Function Documentation

**7.62.1.1 `NppStatus nppiFilter_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, const `Npp32s *pKernel`, `NppiSize oKernelSize`, `NppiPoint oAnchor`, `Npp32s nDivisor`)**

8-bit unsigned convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by `nDivisor`.

#### Parameters:

*`pSrc`* Source-Image Pointer.

*`nSrcStep`* Source-Image Line Step.

*`pDst`* Destination-Image Pointer.

*`nDstStep`* Destination-Image Line Step.

*`oSizeROI`* Region-of-Interest (ROI).

*`pKernel`* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*`oKernelSize`* Width and Height of the rectangular kernel.

*`oAnchor`* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

*`nDivisor`* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.62.1.2 NppStatus nppiFilter\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s \* pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)**

4 channel 8-bit unsigned convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.63 2D Fixed Linear Filters

### 2D Linear Fixed Filters

2D linear fixed filters for 8 bit images.

- `NppStatus nppiFilterBox_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*8-bit unsigned box filter.*
- `NppStatus nppiFilterBox_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*4 channel 8-bit unsigned box filter.*

#### 7.63.1 Function Documentation

##### 7.63.1.1 `NppStatus nppiFilterBox_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)

8-bit unsigned box filter.

Computes the average pixel values of the pixels under a rectangular mask.

##### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- oMaskSize* Width and Height of the neighborhood region for the local Avg operation.
- oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

##### 7.63.1.2 `NppStatus nppiFilterBox_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)

4 channel 8-bit unsigned box filter.

Computes the average pixel values of the pixels under a rectangular mask.

##### Parameters:

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.64 Rank Filters

### Image Rank Filters

Min, Median, and Max image filters.

- `NppStatus nppiFilterMax_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*8-bit unsigned maximum filter.*
- `NppStatus nppiFilterMax_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*4 channel 8-bit unsigned maximum filter.*
- `NppStatus nppiFilterMin_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*8-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)  
*4 channel 8-bit unsigned minimum filter.*

#### 7.64.1 Function Documentation

##### 7.64.1.1 `NppStatus nppiFilterMax_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiSize oMaskSize`, `NppiPoint oAnchor`)

8-bit unsigned maximum filter.

Result pixel value is the maximum of pixel values under the rectangular mask region.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.2 NppStatus nppiFilterMax\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)**

4 channel 8-bit unsigned maximum filter.

Result pixel value is the maximum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.3 NppStatus nppiFilterMin\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)**

8-bit unsigned minimum filter.

Result pixel value is the minimum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.4 NppStatus nppiFilterMin\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)**

4 channel 8-bit unsigned minimum filter.

Result pixel value is the minimum of pixel values under the rectangular mask region.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.65 Geometry Transforms

Routines manipulating an image's geometry.

### Modules

- [Resize](#)
- [Rotate](#)

*Rotates an image around the origin (0,0) and then shifts it.*

- [Mirror](#)
- [Affine Transforms](#)
- [Perspective Transform](#)

### 7.65.1 Detailed Description

Routines manipulating an image's geometry.

### 7.65.2 Geometric Transform API Specifics

This section covers some of the unique API features common to the geometric transform primitives.

#### 7.65.2.1 Geometric Transforms and ROIs

Geometric transforms operate on source and destination ROIs. The way these ROIs affect the processing of pixels differs from other (non geometric) image-processing primitives: Only pixels in the intersection of the destination ROI and the transformed source ROI are being processed.

The typical processing proceeds as follows:

1. Transform the rectangular source ROI (given in source image coordinates) into the destination image space. This yields a quadrilateral.
2. Write only pixels in the intersection of the transformed source ROI and the destination ROI.

#### 7.65.2.2 Pixel Interpolation

The majority of image geometry transform operation need to perform a resampling of the source image as source and destination pixels are not coincident.

NPP supports the following pixel interpolation modes (in order from fastest to slowest and lowest to highest quality):

- nearest neighbor
- linear interpolation
- cubic convolution
- supersampling
- interpolation using Lanczos window function

## 7.66 Resize

### Resize

Resizes 8 bit images.

Handles C1 and C4 images.

- `NppStatus nppiResize_8u_C1R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize dstROISize`, double `xFactor`, double `yFactor`, int `eInterpolation`)  
*8-bit unsigned image resize.*
- `NppStatus nppiResize_8u_C4R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize dstROISize`, double `xFactor`, double `yFactor`, int `eInterpolation`)  
*4 channel 8-bit unsigned image resize.*

### 7.66.1 Detailed Description

### 7.66.2 Error Codes

The resize primitives return the following error codes:

- `NPP_WRONG_INTERSECTION_ROI_ERROR` indicates an error condition if `srcROIRect` has no intersection with the source image.
- `NPP_RESIZE_NO_OPERATION_ERROR` if either destination ROI width or height is less than 1 pixel.
- `NPP_RESIZE_FACTOR_ERROR` Indicates an error condition if either `xFactor` or `yFactor` is less than or equal to zero.
- `NPP_INTERPOLATION_ERROR` if `eInterpolation` has an illegal value.

### 7.66.3 Function Documentation

**7.66.3.1** `NppStatus nppiResize_8u_C1R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiSize dstROISize`, double `xFactor`, double `yFactor`, int `eInterpolation`)

8-bit unsigned image resize.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `oSrcSize` Size in pixels of the source image
- `oSrcROI` Region of interest in the source image.
- `pDst` Destination-Image Pointer.

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image

*xFactor* Factors by which x dimension is changed

*yFactor* Factors by which y dimension is changed

*eInterpolation* The type of eInterpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.66.3.2** `NppStatus nppiResize_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double xFactor, double yFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image

*xFactor* Factors by which x dimension is changed

*yFactor* Factors by which y dimension is changed

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

## 7.67 Rotate

Rotates an image around the origin (0,0) and then shifts it.

### Utility Functions

- `NppStatus nppiGetRotateQuad` (`NppiRect` oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)  
*Compute shape of rotated image.*
- `NppStatus nppiGetRotateBound` (`NppiRect` oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)  
*Compute bounding-box of rotated image.*

### Rotate

- `NppStatus nppiRotate_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*8-bit unsigned image rotate.*
- `NppStatus nppiRotate_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*3 channel 8-bit unsigned image rotate.*
- `NppStatus nppiRotate_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*4 channel 8-bit unsigned image rotate.*
- `NppStatus nppiRotate_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*4 channel 8-bit unsigned image rotate ignoring alpha channel.*
- `NppStatus nppiRotate_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*16-bit unsigned image rotate.*
- `NppStatus nppiRotate_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*3 channel 16-bit unsigned image rotate.*
- `NppStatus nppiRotate_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 16-bit unsigned image rotate.*

- `NppStatus nppiRotate_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, double `nAngle`, double `nShiftX`, double `nShiftY`, int `eInterpolation`)

*4 channel 16-bit unsigned image rotate ignoring alpha channel.*

- `NppStatus nppiRotate_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, double `nAngle`, double `nShiftX`, double `nShiftY`, int `eInterpolation`)

*32-bit float image rotate.*

- `NppStatus nppiRotate_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, double `nAngle`, double `nShiftX`, double `nShiftY`, int `eInterpolation`)

*3 channel 32-bit float image rotate.*

- `NppStatus nppiRotate_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, double `nAngle`, double `nShiftX`, double `nShiftY`, int `eInterpolation`)

*4 channel 32-bit float image rotate.*

- `NppStatus nppiRotate_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, double `nAngle`, double `nShiftX`, double `nShiftY`, int `eInterpolation`)

*4 channel 32-bit float image rotate ignoring alpha channel.*

### 7.67.1 Detailed Description

Rotates an image around the origin (0,0) and then shifts it.

### 7.67.2 Rotate Error Codes

- `NPP_INTERPOLATION_ERROR` if `eInterpolation` has an illegal value.
- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1.
- `NPP_WRONG_INTERSECTION_ROI_ERROR` indicates an error condition if `srcROIrect` has no intersection with the source image.
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` indicates a warning that no operation is performed if the transformed source ROI does not intersect the destination ROI.

### 7.67.3 Function Documentation

#### 7.67.3.1 `NppStatus nppiGetRotateBound` (`NppiRect` `oSrcROI`, `double` `aBoundingBox[2][2]`, `double` `nAngle`, `double` `nShiftX`, `double` `nShiftY`)

Compute bounding-box of rotated image.

**Parameters:**

- oSrcROI* Region-of-interest of the source image.
- aBoundingBox* Two 2D points representing the bounding-box of the rotated image. All four points from `nppiGetRotateQuad` are contained inside the axis-aligned rectangle spanned by the two points of this bounding box.
- nAngle* The rotation angle.
- nShiftX* Post-rotation shift in x-direction.
- nShiftY* Post-rotation shift in y-direction.

**Returns:**

[ROI Related Error Codes.](#)

### 7.67.3.2 `NppStatus nppiGetRotateQuad (NppiRect oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)`

Compute shape of rotated image.

**Parameters:**

- oSrcROI* Region-of-interest of the source image.
- aQuad* Array of 2D points. These points are the locations of the corners of the rotated ROI.
- nAngle* The rotation `nAngle`.
- nShiftX* Post-rotation shift in x-direction
- nShiftY* Post-rotation shift in y-direction

**Returns:**

[ROI Related Error Codes.](#)

### 7.67.3.3 `NppStatus nppiRotate_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate ignoring alpha channel.

**Parameters:**

- pSrc* [Source-Image Pointer.](#)
- nSrcStep* [Source-Image Line Step.](#)
- oSrcSize* Size in pixels of the source image
- oSrcROI* Region of interest in the source image.
- pDst* [Destination-Image Pointer.](#)
- nDstStep* [Destination-Image Line Step.](#)
- oDstROI* Region of interest in the destination image.
- nAngle* The angle of rotation in degrees.
- nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.4** `NppStatus nppiRotate_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

16-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.5** `NppStatus nppiRotate_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 16-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

### 7.67.3.6 NppStatus nppiRotate\_16u\_C4R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 16-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

### 7.67.3.7 NppStatus nppiRotate\_32f\_AC4R (const Npp32f \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f \* pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

4 channel 32-bit float image rotate ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

### 7.67.3.8 NppStatus nppiRotate\_32f\_C1R (const Npp32f \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f \* pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

### 7.67.3.9 NppStatus nppiRotate\_32f\_C3R (const Npp32f \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f \* pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

3 channel 32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.10** `NppStatus nppiRotate_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.11** `NppStatus nppiRotate_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.12** `NppStatus nppiRotate_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.13** `NppStatus nppiRotate_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.67.3.14** `NppStatus nppiRotate_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

## 7.68 Mirror

### Mirror

Mirrors images horizontally, vertically and diagonally.

- `NppStatus nppiMirror_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*8-bit unsigned image mirror.*
- `NppStatus nppiMirror_8u_C3R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*3 channel 8-bit unsigned image mirror.*
- `NppStatus nppiMirror_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*4 channel 8-bit unsigned image mirror.*
- `NppStatus nppiMirror_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*4 channel 8-bit unsigned image mirror not affecting alpha.*
- `NppStatus nppiMirror_16u_C1R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_C3R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*3 channel 16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*4 channel 16-bit unsigned image mirror.*
- `NppStatus nppiMirror_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*4 channel 16-bit unsigned image mirror not affecting alpha.*
- `NppStatus nppiMirror_32s_C1R` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*32-bit image mirror.*
- `NppStatus nppiMirror_32s_C3R` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*3 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_C4R` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize` `oROI`, `NppiAxis` `flip`)  
*4 channel 32-bit image mirror.*

- `NppStatus nppiMirror_32s_AC4R` (const `Npp32s *pSrc`, int `nSrcStep`, `Npp32s *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)  
*4 channel 32-bit image mirror not affecting alpha.*
- `NppStatus nppiMirror_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)  
*32-bit float image mirror.*
- `NppStatus nppiMirror_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)  
*3 channel 32-bit float image mirror.*
- `NppStatus nppiMirror_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)  
*4 channel 32-bit float image mirror.*
- `NppStatus nppiMirror_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)  
*4 channel 32-bit float image mirror not affecting alpha.*

### 7.68.1 Detailed Description

### 7.68.2 Mirror Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

### 7.68.3 Function Documentation

#### 7.68.3.1 `NppStatus nppiMirror_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oROI`, `NppiAxis flip`)

4 channel 16-bit unsigned image mirror not affecting alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

**7.68.3.2 NppStatus nppiMirror\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.3 NppStatus nppiMirror\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.4 NppStatus nppiMirror\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.5 NppStatus nppiMirror\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit float image mirror not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.6 NppStatus nppiMirror\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

32-bit float image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.7 NppStatus nppiMirror\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 32-bit float image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.8 NppStatus nppiMirror\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit float image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.9 NppStatus nppiMirror\_32s\_AC4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit image mirror not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.10 NppStatus nppiMirror\_32s\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

32-bit image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.11 NppStatus nppiMirror\_32s\_C3R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 32-bit image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.12 NppStatus nppiMirror\_32s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 32-bit image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.13 NppStatus nppiMirror\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 8-bit unsigned image mirror not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.14 NppStatus nppiMirror\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

8-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.15 NppStatus nppiMirror\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 8-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.68.3.16 NppStatus nppiMirror\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 8-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

## 7.69 Affine Transforms

### Utility Functions

- `NppStatus nppiGetAffineTransform` (`NppiRect` oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])  
*Computes affine transform coefficients based on source ROI and destination quadrilateral.*
- `NppStatus nppiGetAffineQuad` (`NppiRect` oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])  
*Compute shape of transformed image.*
- `NppStatus nppiGetAffineBound` (`NppiRect` oSrcROI, double aBound[2][2], const double aCoeffs[2][3])  
*Compute bounding-box of transformed image.*

### Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a  $2 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates are computed as follows:

$$x' = c_{00} * x + c_{01} * y + c_{02} \quad y' = c_{10} * x + c_{11} * y + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

Affine transforms can be understood as a linear transformation (traditional matrix multiplication) and a shift operation. The  $2 \times 2$  matrix

$$L = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$$

represents the linear transform portion of the affine transformation. The vector

$$v = \begin{pmatrix} c_{02} \\ c_{12} \end{pmatrix}$$

represents the post-transform shift, i.e. after the pixel location is transformed by  $L$  it is translated by  $v$ .

- `NppStatus nppiWarpAffine_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Single-channel 8-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Three-channel 8-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel 8-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Four-channel 8-bit unsigned affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffine_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[3]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Three-channel planar 8-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[4]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Four-channel planar 8-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_16u_C1R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Single-channel 16-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_16u_C3R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Three-channel 16-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_16u_C4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Four-channel 16-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Four-channel 16-bit unsigned affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffine_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[3]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Three-channel planar 16-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[4]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Four-channel planar 16-bit unsigned affine warp.*
- `NppStatus nppiWarpAffine_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)  
*Single-channel 32-bit signed affine warp.*

- `NppStatus nppiWarpAffine_32s_C3R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Three-channel 32-bit signed affine warp.*
- `NppStatus nppiWarpAffine_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Four-channel 32-bit signed affine warp.*
- `NppStatus nppiWarpAffine_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Four-channel 32-bit signed affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffine_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[3]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Three-channel planar 32-bit signed affine warp.*
- `NppStatus nppiWarpAffine_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[4]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Four-channel planar 32-bit signed affine warp.*
- `NppStatus nppiWarpAffine_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Single-channel 32-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Three-channel 32-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Four-channel 32-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Four-channel 32-bit floating-point affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffine_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` `oSrcSize`, `int` `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst[3]`, `int` `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, `int` `eInterpolation`)

*Three-channel planar 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel planar 32-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_64f_C1R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Single-channel 64-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_64f_C3R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Three-channel 64-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_64f_C4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel 64-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_64f_AC4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel 64-bit floating-point affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffine_64f_P3R` (const `Npp64f *aSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[3]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Three-channel planar 64-bit floating-point affine warp.*
- `NppStatus nppiWarpAffine_64f_P4R` (const `Npp64f *aSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel planar 64-bit floating-point affine warp.*

## Backwards Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a  $2 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates fulfill the following properties:

$$x = c_{00} * x' + c_{01} * y' + c_{02} \quad y = c_{10} * x' + c_{11} * y' + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

In other words, given matrix  $C$  the source image's shape is transformed to the destination image using the inverse matrix  $C^{-1}$ :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix} \quad x' = m_{00} * x + m_{01} * y + m_{02} \quad y' = m_{10} * x + m_{11} * y + m_{12}$$

- `NppStatus nppiWarpAffineBack_8u_C1R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 8-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_8u_C3R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel 8-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_8u_C4R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 8-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineBack_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel planar 8-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel planar 8-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_16u_C1R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 16-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_16u_C3R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel 16-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_16u_C4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineBack_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel planar 16-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel planar 16-bit unsigned integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 32-bit signed integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32s_C3R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel 32-bit signed integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineBack_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel planar 32-bit signed integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel planar 32-bit signed integer backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 32-bit floating-point backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel 32-bit floating-point backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel 32-bit floating-point backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineBack_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst[3]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Three-channel planar 32-bit floating-point backwards affine warp.*
- `NppStatus nppiWarpAffineBack_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)  
*Four-channel planar 32-bit floating-point backwards affine warp.*

## Quad-Based Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

An affine transform is fully determined by the mapping of 3 discrete points. The following primitives compute an affine transformation matrix that maps the first three corners of the source quad are mapped to the first three vertices of the destination image quad. If the fourth vertices do not match the transform, an `NPP_AFFINE_QUAD_INCORRECT_WARNING` is returned by the primitive.

- `NppStatus nppiWarpAffineQuad_8u_C1R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Single-channel 32-bit floating-point quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_8u_C3R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Three-channel 8-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_8u_C4R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 8-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineQuad_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Three-channel planar 8-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Four-channel planar 8-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_16u_C1R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Single-channel 16-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_16u_C3R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Three-channel 16-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_16u_C4R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Four-channel 16-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineQuad_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Three-channel planar 16-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp16u *pDst[4]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Four-channel planar 16-bit unsigned integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32s_C1R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Single-channel 32-bit signed integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32s_C3R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)  
*Three-channel 32-bit signed integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel 32-bit signed integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineQuad_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Three-channel planar 32-bit signed integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32s *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel planar 32-bit signed integer quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Single-channel 32-bit floating-point quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Three-channel 32-bit floating-point quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel 32-bit floating-point quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.*
- `NppStatus nppiWarpAffineQuad_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Three-channel planar 32-bit floating-point quad-based affine warp.*
- `NppStatus nppiWarpAffineQuad_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel planar 32-bit floating-point quad-based affine warp.*

### 7.69.1 Detailed Description

### 7.69.2 Affine Transform Error Codes

- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) Indicates an error condition if oSrcROI has no intersection with the source image
- [NPP\\_INTERPOLATION\\_ERROR](#) Indicates an error condition if interpolation has an illegal value
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_WRONG\\_INTERSECTION\\_QUAD\\_WARNING](#) Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

### 7.69.3 Function Documentation

#### 7.69.3.1 `NppStatus nppiGetAffineBound (NppiRect oSrcROI, double aBound[2][2], const double aCoeffs[2][3])`

Compute bounding-box of transformed image.

The method effectively computes the bounding box (axis aligned rectangle) of the transformed source ROI (see [nppiGetAffineQuad\(\)](#)).

#### Parameters:

- oSrcROI* The source ROI.
- aBound* The resulting bounding box.
- aCoeffs* The affine transform coefficients.

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

#### 7.69.3.2 `NppStatus nppiGetAffineQuad (NppiRect oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])`

Compute shape of transformed image.

This method computes the quadrilateral in the destination image that the source ROI is transformed into by the affine transformation expressed by the coefficients array (*aCoeffs*).

#### Parameters:

- oSrcROI* The source ROI.

*aQuad* The resulting destination quadrangle.

*aCoeffs* The affine transform coefficients.

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

**7.69.3.3 NppStatus nppiGetAffineTransform (NppiRect oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])**

Computes affine transform coefficients based on source ROI and destination quadrilateral.

The function computes the coefficients of an affine transformation that maps the given source ROI (axis aligned rectangle with integer coordinates) to a quadrilateral in the destination image.

An affine transform in 2D is fully determined by the mapping of just three vertices. This function's API allows for passing a complete quadrilateral effectively making the problem overdetermined. What this means in practice is, that for certain quadrilaterals it is not possible to find an affine transform that would map all four corners of the source ROI to the four vertices of that quadrilateral.

The function circumvents this problem by only looking at the first three vertices of the destination image quadrilateral to determine the affine transformation's coefficients. If the destination quadrilateral is indeed one that cannot be mapped using an affine transformation the function informs the user of this situation by returning a [NPP\\_AFFINE\\_QUAD\\_INCORRECT\\_WARNING](#).

**Parameters:**

*oSrcROI* The source ROI. This rectangle needs to be at least one pixel wide and high. If either width or height are less than one an [NPP\\_RECT\\_ERROR](#) is returned.

*aQuad* The destination quadrilateral.

*aCoeffs* The resulting affine transform coefficients.

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_AFFINE\\_QUAD\\_INCORRECT\\_WARNING](#) Indicates a warning when quad does not conform to the transform properties. Fourth vertex is ignored, internally computed coordinates are used instead

**7.69.3.4 NppStatus nppiWarpAffine\_16u\_AC4R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)**

Four-channel 16-bit unsigned affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.5 NppStatus nppiWarpAffine\_16u\_C1R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)**

Single-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.6 NppStatus nppiWarpAffine\_16u\_C3R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)**

Three-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.7 NppStatus nppiWarpAffine\_16u\_C4R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)**

Four-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.8** `NppStatus nppiWarpAffine_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.9** `NppStatus nppiWarpAffine_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.10** `NppStatus nppiWarpAffine_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.11** `NppStatus nppiWarpAffine_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.12** `NppStatus nppiWarpAffine_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.13** `NppStatus nppiWarpAffine_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.14** `NppStatus nppiWarpAffine_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.15** `NppStatus nppiWarpAffine_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.16** `NppStatus nppiWarpAffine_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.17** `NppStatus nppiWarpAffine_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.18** `NppStatus nppiWarpAffine_32s_C3R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect oSrcROI`, `Npp32s * pDst`, `int nDstStep`, `NppiRect oDstROI`, `const double aCoeffs[2][3]`, `int eInterpolation`)

Three-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.19** `NppStatus nppiWarpAffine_32s_C4R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect oSrcROI`, `Npp32s * pDst`, `int nDstStep`, `NppiRect oDstROI`, `const double aCoeffs[2][3]`, `int eInterpolation`)

Four-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.20** `NppStatus nppiWarpAffine_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.21** `NppStatus nppiWarpAffine_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.22** `NppStatus nppiWarpAffine_64f_AC4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.23** `NppStatus nppiWarpAffine_64f_C1R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.24** `NppStatus nppiWarpAffine_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.25** `NppStatus nppiWarpAffine_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.26** `NppStatus nppiWarpAffine_64f_P3R (const Npp64f * aSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 64-bit floating-point affine warp.

**Parameters:**

*aSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.27** `NppStatus nppiWarpAffine_64f_P4R (const Npp64f * aSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 64-bit floating-point affine warp.

**Parameters:**

*aSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.28** `NppStatus nppiWarpAffine_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.29** `NppStatus nppiWarpAffine_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.30** `NppStatus nppiWarpAffine_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.31** `NppStatus nppiWarpAffine_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.32** `NppStatus nppiWarpAffine_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.33** `NppStatus nppiWarpAffine_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.34** `NppStatus nppiWarpAffineBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.35** `NppStatus nppiWarpAffineBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.36** `NppStatus nppiWarpAffineBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.37** `NppStatus nppiWarpAffineBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.38** `NppStatus nppiWarpAffineBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.39** `NppStatus nppiWarpAffineBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.40** `NppStatus nppiWarpAffineBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.41** `NppStatus nppiWarpAffineBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.42** `NppStatus nppiWarpAffineBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.43** `NppStatus nppiWarpAffineBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.44** `NppStatus nppiWarpAffineBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.45** `NppStatus nppiWarpAffineBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.46** `NppStatus nppiWarpAffineBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.47** `NppStatus nppiWarpAffineBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.48** `NppStatus nppiWarpAffineBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.49** `NppStatus nppiWarpAffineBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.50** `NppStatus nppiWarpAffineBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.51** `NppStatus nppiWarpAffineBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.52** `NppStatus nppiWarpAffineBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.53** `NppStatus nppiWarpAffineBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.54** `NppStatus nppiWarpAffineBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.55** `NppStatus nppiWarpAffineBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.56** `NppStatus nppiWarpAffineBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.57** `NppStatus nppiWarpAffineBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.58** `NppStatus nppiWarpAffineQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.59** `NppStatus nppiWarpAffineQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.60** `NppStatus nppiWarpAffineQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.61** `NppStatus nppiWarpAffineQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.62** `NppStatus nppiWarpAffineQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.63** `NppStatus nppiWarpAffineQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.64** `NppStatus nppiWarpAffineQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.65** `NppStatus nppiWarpAffineQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.66** `NppStatus nppiWarpAffineQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.67** `NppStatus nppiWarpAffineQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.68** `NppStatus nppiWarpAffineQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.69** `NppStatus nppiWarpAffineQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.70** `NppStatus nppiWarpAffineQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.71** `NppStatus nppiWarpAffineQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.72** `NppStatus nppiWarpAffineQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.73** `NppStatus nppiWarpAffineQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.74** `NppStatus nppiWarpAffineQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.75** `NppStatus nppiWarpAffineQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.76** `NppStatus nppiWarpAffineQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.77** `NppStatus nppiWarpAffineQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.78** `NppStatus nppiWarpAffineQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.79** `NppStatus nppiWarpAffineQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.80** `NppStatus nppiWarpAffineQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.69.3.81** `NppStatus nppiWarpAffineQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

## 7.70 Perspective Transform

### Utility Functions

- [NppStatus nppiGetPerspectiveTransform](#) ([NppiRect](#) oSrcROI, const double quad[4][2], double aCoeffs[3][3])  
*Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.*
- [NppStatus nppiGetPerspectiveQuad](#) ([NppiRect](#) oSrcROI, double quad[4][2], const double aCoeffs[3][3])  
*Calculates perspective transform projection of given source rectangular ROI.*
- [NppStatus nppiGetPerspectiveBound](#) ([NppiRect](#) oSrcROI, double bound[2][2], const double aCoeffs[3][3])  
*Calculates bounding box of the perspective transform projection of the given source rectangular ROI.*

### Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a  $3 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates are computed as follows:

$$x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

- [NppStatus nppiWarpPerspective\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Single-channel 8-bit unsigned integer perspective warp.*
- [NppStatus nppiWarpPerspective\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel 8-bit unsigned integer perspective warp.*
- [NppStatus nppiWarpPerspective\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 8-bit unsigned integer perspective warp.*
- [NppStatus nppiWarpPerspective\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.*

- **NppStatus** **nppiWarpPerspective\_8u\_P3R** (const **Npp8u** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst[3], int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel planar 8-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_8u\_P4R** (const **Npp8u** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst[4], int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel planar 8-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_16u\_C1R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Single-channel 16-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_16u\_C3R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel 16-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_16u\_C4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 16-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_16u\_AC4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.*
- **NppStatus** **nppiWarpPerspective\_16u\_P3R** (const **Npp16u** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst[3], int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel planar 16-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_16u\_P4R** (const **Npp16u** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst[4], int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel planar 16-bit unsigned integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_32s\_C1R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32s** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Single-channel 32-bit signed integer perspective warp.*
- **NppStatus** **nppiWarpPerspective\_32s\_C3R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32s** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer perspective warp.*
- `NppStatus nppiWarpPerspective_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.*
- `NppStatus nppiWarpPerspective_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel planar 32-bit signed integer perspective warp.*
- `NppStatus nppiWarpPerspective_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel planar 32-bit signed integer perspective warp.*
- `NppStatus nppiWarpPerspective_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Single-channel 32-bit floating-point perspective warp.*
- `NppStatus nppiWarpPerspective_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel 32-bit floating-point perspective warp.*
- `NppStatus nppiWarpPerspective_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit floating-point perspective warp.*
- `NppStatus nppiWarpPerspective_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.*
- `NppStatus nppiWarpPerspective_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel planar 32-bit floating-point perspective warp.*
- `NppStatus nppiWarpPerspective_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32f *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel planar 32-bit floating-point perspective warp.*

## Backwards Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a  $3 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates fulfill the following properties:

$$x = \frac{c_{00} * x' + c_{01} * y' + c_{02}}{c_{20} * x' + c_{21} * y' + c_{22}} \quad y = \frac{c_{10} * x' + c_{11} * y' + c_{12}}{c_{20} * x' + c_{21} * y' + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

In other words, given matrix  $C$  the source image's shape is transformed to the destination image using the inverse matrix  $C^{-1}$ :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \quad x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

- `NppStatus nppiWarpPerspectiveBack_8u_C1R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Single-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_C3R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_C4R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel planar 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel planar 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C1R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Single-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C3R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C4R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel planar 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp16u *pDst[4]`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel planar 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C1R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Single-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C3R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C4R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, `Npp32s *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aCoeffs[3][3]`, int `eInterpolation`)

*Three-channel planar 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel planar 32-bit signed integer backwards perspective warp.*
- `NppStatus nppiWarpPerspectiveBack_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Single-channel 32-bit floating-point backwards perspective warp.*
- `NppStatus nppiWarpPerspectiveBack_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel 32-bit floating-point backwards perspective warp.*
- `NppStatus nppiWarpPerspectiveBack_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 32-bit floating-point backwards perspective warp.*
- `NppStatus nppiWarpPerspectiveBack_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.*
- `NppStatus nppiWarpPerspectiveBack_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Three-channel planar 32-bit floating-point backwards perspective warp.*
- `NppStatus nppiWarpPerspectiveBack_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)  
*Four-channel planar 32-bit floating-point backwards perspective warp.*

## Quad-Based Perspective Transform

Transforms (warps) an image based on an perspective transform.

The perspective transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

- `NppStatus nppiWarpPerspectiveQuad_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Single-channel 8-bit unsigned integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Single-channel 32-bit signed integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32s_C3R` (const `Npp32s *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Three-channel 32-bit signed integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32s_C4R` (const `Npp32s *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 32-bit signed integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32s_AC4R` (const `Npp32s *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.*
- `NppStatus nppiWarpPerspectiveQuad_32s_P3R` (const `Npp32s *pSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst[3]`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Three-channel planar 32-bit signed integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32s_P4R` (const `Npp32s *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s *pDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel planar 32-bit signed integer quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32f_C1R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Single-channel 32-bit floating-point quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32f_C3R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Three-channel 32-bit floating-point quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32f_C4R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 32-bit floating-point quad-based perspective warp.*
- `NppStatus nppiWarpPerspectiveQuad_32f_AC4R` (const `Npp32f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)  
*Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[3]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Three-channel planar 32-bit floating-point quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize oSrcSize`, int `nSrcStep`, `NppiRect oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[4]`, int `nDstStep`, `NppiRect oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel planar 32-bit floating-point quad-based perspective warp.*

### 7.70.1 Detailed Description

#### 7.70.2 Perspective Transform Error Codes

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `oSrcROI` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if interpolation has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

#### 7.70.3 Function Documentation

##### 7.70.3.1 `NppStatus nppiGetPerspectiveBound (NppiRect oSrcROI, double bound[2][2], const double aCoeffs[3][3])`

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

##### Parameters:

*oSrcROI* Source ROI

*bound* Bounding box of the transformed source ROI

*aCoeffs* Perspective transform coefficients

##### Returns:

Error codes:

- `NPP_SIZE_ERROR` Indicates an error condition if any image dimension has zero or negative value
- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

### 7.70.3.2 NppStatus nppiGetPerspectiveQuad (NppiRect *oSrcROI*, double *quad*[4][2], const double *aCoeffs*[3][3])

Calculates perspective transform projection of given source rectangular ROI.

#### Parameters:

*oSrcROI* Source ROI  
*quad* Destination quadrangle  
*aCoeffs* Perspective transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

### 7.70.3.3 NppStatus nppiGetPerspectiveTransform (NppiRect *oSrcROI*, const double *quad*[4][2], double *aCoeffs*[3][3])

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

#### Parameters:

*oSrcROI* Source ROI  
*quad* Destination quadrangle  
*aCoeffs* Perspective transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

### 7.70.3.4 NppStatus nppiWarpPerspective\_16u\_AC4R (const Npp16u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, const double *aCoeffs*[3][3], int *eInterpolation*)

Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcROI* Source ROI  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

### 7.70.3.5 NppStatus nppiWarpPerspective\_16u\_C1R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)

Single-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcROI* Source ROI  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

### 7.70.3.6 NppStatus nppiWarpPerspective\_16u\_C3R (const Npp16u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)

Three-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.  
*oSrcROI* Source ROI  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.70.3.7** `NppStatus nppiWarpPerspective_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.  
*oSrcSize* Size of source image in pixels  
*nSrcStep* Source-Image Line Step.  
*oSrcROI* Source ROI  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.70.3.8** `NppStatus nppiWarpPerspective_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.  
*oSrcSize* Size of source image in pixels  
*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.70.3.9** `NppStatus nppiWarpPerspective_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.  
*oSrcSize* Size of source image in pixels  
*nSrcStep* Source-Image Line Step.  
*oSrcROI* Source ROI  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.70.3.10** `NppStatus nppiWarpPerspective_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*oSrcSize* Size of source image in pixels  
*nSrcStep* Source-Image Line Step.  
*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.11** `NppStatus nppiWarpPerspective_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.12** `NppStatus nppiWarpPerspective_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.13** `NppStatus nppiWarpPerspective_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.14** `NppStatus nppiWarpPerspective_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.15** `NppStatus nppiWarpPerspective_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.16** `NppStatus nppiWarpPerspective_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.17** `NppStatus nppiWarpPerspective_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.18** `NppStatus nppiWarpPerspective_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.19** `NppStatus nppiWarpPerspective_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.20** `NppStatus nppiWarpPerspective_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.21 NppStatus nppiWarpPerspective\_32s\_P4R (const Npp32s \* pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s \* pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)**

Four-channel planar 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.22 NppStatus nppiWarpPerspective\_8u\_AC4R (const Npp8u \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)**

Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.23** `NppStatus nppiWarpPerspective_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.24** `NppStatus nppiWarpPerspective_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.25** `NppStatus nppiWarpPerspective_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.26** `NppStatus nppiWarpPerspective_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.27** `NppStatus nppiWarpPerspective_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.28** `NppStatus nppiWarpPerspectiveBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.29** `NppStatus nppiWarpPerspectiveBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.30** `NppStatus nppiWarpPerspectiveBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.31** `NppStatus nppiWarpPerspectiveBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.32** `NppStatus nppiWarpPerspectiveBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.33** `NppStatus nppiWarpPerspectiveBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.34** `NppStatus nppiWarpPerspectiveBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.335** `NppStatus nppiWarpPerspectiveBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.336** `NppStatus nppiWarpPerspectiveBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.37** `NppStatus nppiWarpPerspectiveBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.38** `NppStatus nppiWarpPerspectiveBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.39** `NppStatus nppiWarpPerspectiveBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.40** `NppStatus nppiWarpPerspectiveBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.41 NppStatus nppiWarpPerspectiveBack\_32s\_C1R (const Npp32s \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)**

Single-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.42 NppStatus nppiWarpPerspectiveBack\_32s\_C3R (const Npp32s \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s \* pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)**

Three-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.43** `NppStatus nppiWarpPerspectiveBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.44** `NppStatus nppiWarpPerspectiveBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.45** `NppStatus nppiWarpPerspectiveBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.46** `NppStatus nppiWarpPerspectiveBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.47** `NppStatus nppiWarpPerspectiveBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.48** `NppStatus nppiWarpPerspectiveBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.49** `NppStatus nppiWarpPerspectiveBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.50** `NppStatus nppiWarpPerspectiveBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.51** `NppStatus nppiWarpPerspectiveBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.52** `NppStatus nppiWarpPerspectiveQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.53** `NppStatus nppiWarpPerspectiveQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.54** `NppStatus nppiWarpPerspectiveQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.55** `NppStatus nppiWarpPerspectiveQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.56** `NppStatus nppiWarpPerspectiveQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.57** `NppStatus nppiWarpPerspectiveQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.58** `NppStatus nppiWarpPerspectiveQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.59** `NppStatus nppiWarpPerspectiveQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.60** `NppStatus nppiWarpPerspectiveQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.61** `NppStatus nppiWarpPerspectiveQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.62** `NppStatus nppiWarpPerspectiveQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.63** `NppStatus nppiWarpPerspectiveQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.64** `NppStatus nppiWarpPerspectiveQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.65** `NppStatus nppiWarpPerspectiveQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.66** `NppStatus nppiWarpPerspectiveQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.67** `NppStatus nppiWarpPerspectiveQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.68** `NppStatus nppiWarpPerspectiveQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.69** `NppStatus nppiWarpPerspectiveQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.70** `NppStatus nppiWarpPerspectiveQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.71** `NppStatus nppiWarpPerspectiveQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.72** `NppStatus nppiWarpPerspectiveQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.73** `NppStatus nppiWarpPerspectiveQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.74** `NppStatus nppiWarpPerspectiveQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.70.3.75** `NppStatus nppiWarpPerspectiveQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

## 7.71 Linear Transforms

Linear image transformations.

### Modules

- [Fourier Transforms](#)

### 7.71.1 Detailed Description

Linear image transformations.

## 7.72 Fourier Transforms

### Functions

- `NppStatus nppiMagnitude_32fc32f_C1R` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*32-bit floating point complex to 32-bit floating point magnitude.*
- `NppStatus nppiMagnitudeSqr_32fc32f_C1R` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)  
*32-bit floating point complex to 32-bit floating point squared magnitude.*

### 7.72.1 Function Documentation

#### 7.72.1.1 `NppStatus nppiMagnitude_32fc32f_C1R` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

32-bit floating point complex to 32-bit floating point magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the magnitude of the complex values.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.
- `nDstStep` Destination-Image Line Step.
- `oSizeROI` Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.72.1.2 `NppStatus nppiMagnitudeSqr_32fc32f_C1R` (const `Npp32fc *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

32-bit floating point complex to 32-bit floating point squared magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the squared magnitude of the complex values.

The squared magnitude is an intermediate result of magnitude computation and can thus be computed faster than actual magnitude. If magnitudes are required for sorting/comparing only, using this function instead of `nppiMagnitude_32fc32f_C1R` can be a worthwhile performance optimization.

#### Parameters:

- `pSrc` Source-Image Pointer.
- `nSrcStep` Source-Image Line Step.
- `pDst` Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.73 Morphological Operations

Morphological image operations.

### Modules

- [Dilation And Erosion](#)

### 7.73.1 Detailed Description

Morphological image operations.

## 7.74 Dilation And Erosion

### Functions

- [NppStatus nppiDilate\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)  
*8-bit unsigned image dilation.*
- [NppStatus nppiDilate\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)  
*4 channel 8-bit unsigned image dilation.*
- [NppStatus nppiErode\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)  
*8-bit unsigned image erosion.*
- [NppStatus nppiErode\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)  
*4 channel 8-bit unsigned image erosion.*

### 7.74.1 Function Documentation

#### 7.74.1.1 [NppStatus nppiDilate\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) \*pMask, [NppiSize](#) oMaskSize, [NppiPoint](#) oAnchor)

8-bit unsigned image dilation.

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.74.1.2 NppStatus nppiDilate\_8u\_C4R** (**const Npp8u \* pSrc**, **int nSrcStep**, **Npp8u \* pDst**, **int nDstStep**, **NppiSize oSizeROI**, **const Npp8u \* pMask**, **NppiSize oMaskSize**, **NppiPoint oAnchor**)

4 channel 8-bit unsigned image dilation.

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.74.1.3 NppStatus nppiErode\_8u\_C1R** (**const Npp8u \* pSrc**, **Npp32s nSrcStep**, **Npp8u \* pDst**, **Npp32s nDstStep**, **NppiSize oSizeROI**, **const Npp8u \* pMask**, **NppiSize oMaskSize**, **NppiPoint oAnchor**)

8-bit unsigned image erosion.

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.74.1.4 NppStatus nppiErode\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, NppiSize oMaskSize, NppiPoint oAnchor)**

4 channel 8-bit unsigned image erosion.

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask. Pixels whose corresponding mask values are zero do not participate in the maximum search.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.75 Statistics Functions

Routines computing statistical image information.

### Modules

- [Sum](#)
- [Minimum](#)
- [Maximum](#)
- [Minimum\\_Maximum](#)
- [Mean](#)
- [Mean And Standard Deviation](#)
- [Infinity Norm](#)
- [L1 Norm](#)
- [L2 Norm](#)
- [Norm Diff](#)
- [Integral and Rectangular Standard Deviation](#)
- [Histogram](#)

### 7.75.1 Detailed Description

Routines computing statistical image information.

## 7.76 Sum

### Sum

Sum functions compute the sum of all the pixel values in an image.

If the image contains multiple channels, the sums will be calculated for each channel separately. Functions also require scratch buffer during the computation. For details, please refer [Scratch Buffer and Host Pointer](#). The `nppiSumGetBuffer_X_X` functions compute the size of the scratch buffer. It is the user's responsibility to allocate the sufficient GPU memory based on the size and pass the memory pointer to the sum functions.

- `NppStatus nppiSumGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_8u_C1R`.*
- `NppStatus nppiSumGetBufferHostSize_8u64s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_8u64s_C1R`.*
- `NppStatus nppiSumGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16u_C1R`.*
- `NppStatus nppiSumGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16s_C1R`.*
- `NppStatus nppiSumGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_32f_C1R`.*
- `NppStatus nppiSumGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_8u_C3R`.*
- `NppStatus nppiSumGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16u_C3R`.*
- `NppStatus nppiSumGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16s_C3R`.*
- `NppStatus nppiSumGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_32f_C3R`.*
- `NppStatus nppiSumGetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_8u_AC4R`.*
- `NppStatus nppiSumGetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16u_AC4R`.*
- `NppStatus nppiSumGetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_16s_AC4R`.*
- `NppStatus nppiSumGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for `nppiSum_32f_AC4R`.*
- `NppStatus nppiSumGetBufferHostSize_8u64s_C4R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)

*Device scratch buffer size (in bytes) for nppiSum\_8u64s\_C4R.*

- **NppStatus nppiSumGetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiSum\_8u\_C4R.*
- **NppStatus nppiSumGetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiSum\_16u\_C4R.*
- **NppStatus nppiSumGetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiSum\_16s\_C4R.*
- **NppStatus nppiSumGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiSum\_32f\_C4R.*
- **NppStatus nppiSum\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pSum)  
*1-channel 8-bit unsigned char image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_8u64s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64s** \*pSum)  
*1-channel 8-bit unsigned char image sum with 64-bit long long result.*
- **NppStatus nppiSum\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pSum)  
*1-channel 16-bit unsigned short image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pSum)  
*1-channel 16-bit signed short image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pSum)  
*1-channel 32-bit floating point image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*3-channel 8-bit unsigned char image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*3-channel 16-bit unsigned short image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*3-channel 16-bit signed short image sum with 64-bit double precision result.*
- **NppStatus nppiSum\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*3-channel 32-bit floating point image sum with 64-bit double precision result.*

- `NppStatus nppiSum_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[3]`)  
*4-channel 8-bit unsigned char image sum with 64-bit double precision result.*
- `NppStatus nppiSum_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[3]`)  
*4-channel 16-bit unsigned short image sum with 64-bit double precision result.*
- `NppStatus nppiSum_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[3]`)  
*4-channel 16-bit signed short image sum with 64-bit double precision result.*
- `NppStatus nppiSum_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[3]`)  
*4-channel 32-bit floating point image sum with 64-bit double precision result.*
- `NppStatus nppiSum_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[4]`)  
*4-channel 8-bit unsigned char image sum with 64-bit double precision result.*
- `NppStatus nppiSum_8u64s_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64s aSum[4]`)  
*4-channel 8-bit unsigned char image sum with 64-bit long long result.*
- `NppStatus nppiSum_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[4]`)  
*4-channel 16-bit unsigned short image sum with 64-bit double precision result.*
- `NppStatus nppiSum_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[4]`)  
*4-channel 16-bit signed short image sum with 64-bit double precision result.*
- `NppStatus nppiSum_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[4]`)  
*4-channel 32-bit floating point image sum with 64-bit double precision result.*

## 7.76.1 Function Documentation

### 7.76.1.1 `NppStatus nppiSum_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f aSum[3]`)

4-channel 16-bit signed short image sum with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.  
*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

#### 7.76.1.2 NppStatus nppiSum\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pSum)

1-channel 16-bit signed short image sum with 64-bit double precision result.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_C1R](#) to determine the minimum number of bytes required.  
*pSum* Pointer to the computed sum.

#### 7.76.1.3 NppStatus nppiSum\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])

3-channel 16-bit signed short image sum with 64-bit double precision result.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_C3R](#) to determine the minimum number of bytes required.  
*aSum* Array that contains computed sum for each channel.

#### 7.76.1.4 NppStatus nppiSum\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[4])

4-channel 16-bit signed short image sum with 64-bit double precision result.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_16s\\_C4R](#) to determine the minimum number of bytes required.  
*aSum* Array that contains computed sum for each channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.5 NppStatus nppiSum\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])**

4-channel 16-bit unsigned short image sum with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16u\\_AC4R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

**7.76.1.6 NppStatus nppiSum\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pSum)**

1-channel 16-bit unsigned short image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16u\\_C1R](#) to determine the minimum number of bytes required.

*pSum* Pointer to the computed sum.

**7.76.1.7 NppStatus nppiSum\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])**

3-channel 16-bit unsigned short image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16u\\_C3R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel.

**7.76.1.8 NppStatus nppiSum\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[4])**

4-channel 16-bit unsigned short image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.9 NppStatus nppiSum\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])**

4-channel 32-bit floating point image sum with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.10 NppStatus nppiSum\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pSum)**

1-channel 32-bit floating point image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.11 NppStatus nppiSum\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])**

3-channel 32-bit floating point image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_32f\\_C3R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel.

**7.76.1.12 NppStatus nppiSum\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[4])**

4-channel 32-bit floating point image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_C4R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.13 NppStatus nppiSum\_8u64s\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64s \* pSum)**

1-channel 8-bit unsigned char image sum with 64-bit long long result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u64s\\_C1R](#) to determine the minimum number of bytes required.

*pSum* Pointer to the computed sum.

**7.76.1.14** `NppStatus nppiSum_8u64s_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64s aSum[4])`

4-channel 8-bit unsigned char image sum with 64-bit long long result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_8u64s\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.15** `NppStatus nppiSum_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

4-channel 8-bit unsigned char image sum with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

**7.76.1.16** `NppStatus nppiSum_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)`

1-channel 8-bit unsigned char image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.17** `NppStatus nppiSum_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

3-channel 8-bit unsigned char image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_C3R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel.

**7.76.1.18** `NppStatus nppiSum_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[4])`

4-channel 8-bit unsigned char image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_C4R](#) to determine the minimum number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.76.1.19** `NppStatus nppiSumGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiSum_16s_AC4R`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.20 NppStatus nppiSumGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.21 NppStatus nppiSumGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.22 NppStatus nppiSumGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.23 NppStatus nppiSumGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.24 NppStatus nppiSumGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.25 NppStatus nppiSumGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.26 NppStatus nppiSumGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_16u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.27 NppStatus nppiSumGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.28 NppStatus nppiSumGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.29 NppStatus nppiSumGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.30 NppStatus nppiSumGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiSum\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.31 NppStatus nppiSumGetBufferHostSize\_8u64s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_8u64s\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.32 NppStatus nppiSumGetBufferHostSize\_8u64s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_8u64s\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.33 NppStatus nppiSumGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiSum\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.34 NppStatus nppiSumGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiSum\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.35 NppStatus nppiSumGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiSum\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.76.1.36 NppStatus nppiSumGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiSum\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.77 Minimum

### Min

These min routines find the minimal pixel value of an image.

If the image has multiple channels, the functions find the minimum for each channel separately. The scratch buffer is also required by the functions.

- [NppStatus nppiMinGetBufferHostSize\\_8u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_8u\_C1R.*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16u\_C1R.*
- [NppStatus nppiMinGetBufferHostSize\\_16s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16s\_C1R.*
- [NppStatus nppiMinGetBufferHostSize\\_32f\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_32f\_C1R.*
- [NppStatus nppiMinGetBufferHostSize\\_8u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_8u\_C3R.*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16u\_C3R.*
- [NppStatus nppiMinGetBufferHostSize\\_16s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16s\_C3R.*
- [NppStatus nppiMinGetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_32f\_C3R.*
- [NppStatus nppiMinGetBufferHostSize\\_8u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_8u\_C4R.*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16u\_C4R.*
- [NppStatus nppiMinGetBufferHostSize\\_16s\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16s\_C4R.*
- [NppStatus nppiMinGetBufferHostSize\\_32f\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_32f\_C4R.*
- [NppStatus nppiMinGetBufferHostSize\\_8u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_8u\_AC4R.*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16u\_AC4R.*

- **NppStatus nppiMinGetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_16s\_AC4R.*
- **NppStatus nppiMinGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMin\_32f\_AC4R.*
- **NppStatus nppiMin\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** \*pMin)  
*1-channel 8-bit unsigned char image min.*
- **NppStatus nppiMin\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** \*pMin)  
*1-channel 16-bit unsigned short integer image min.*
- **NppStatus nppiMin\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** \*pMin)  
*1-channel 16-bit signed short integer image min.*
- **NppStatus nppiMin\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** \*pMin)  
*1-channel 32-bit floating point image min.*
- **NppStatus nppiMin\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[3])  
*3-channel 8-bit unsigned char image min.*
- **NppStatus nppiMin\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[3])  
*3-channel 16-bit unsigned short integer image min.*
- **NppStatus nppiMin\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3])  
*3-channel 16-bit signed short integer image min.*
- **NppStatus nppiMin\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[3])  
*3-channel 32-bit floating point image min.*
- **NppStatus nppiMin\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[4])  
*4-channel 8-bit unsigned char image min.*
- **NppStatus nppiMin\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[4])  
*4-channel 16-bit unsigned short integer image min.*
- **NppStatus nppiMin\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[4])  
*4-channel 16-bit signed short integer image min.*

- `NppStatus nppiMin_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMin[4]`)  
*4-channel 32-bit floating point image min.*
- `NppStatus nppiMin_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMin[3]`)  
*4-channel 8-bit unsigned char image min (alpha channel is not processed).*
- `NppStatus nppiMin_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMin[3]`)  
*4-channel 16-bit unsigned short integer image min (alpha channel is not processed).*
- `NppStatus nppiMin_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMin[3]`)  
*4-channel 16-bit signed short integer image min (alpha channel is not processed).*
- `NppStatus nppiMin_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMin[3]`)  
*4-channel 32-bit floating point image min (alpha channel is not processed).*

## MinIndx

The functions find the minimal value and its indices (X and Y coordinates) of an image.

If the image contains multiple channels, the function will find the values and the indices for each channel separately. If there are several minima in the selected region of interest, the function returns the top leftmost position.

- `NppStatus nppiMinIndxGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C1R.*
- `NppStatus nppiMinIndxGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C1R.*
- `NppStatus nppiMinIndxGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C1R.*
- `NppStatus nppiMinIndxGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_32f\_C1R.*
- `NppStatus nppiMinIndxGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C3R.*
- `NppStatus nppiMinIndxGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C3R.*
- `NppStatus nppiMinIndxGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C3R.*

- **NppStatus nppiMinIdxGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C3R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_8u\_C4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_16u\_C4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_16s\_C4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_8u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_8u\_AC4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_16u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_16u\_AC4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_16s\_AC4R.*
- **NppStatus nppiMinIdxGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_AC4R.*
- **NppStatus nppiMinIdx\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** \*pMin, int \*pIndexX, int \*pIndexY)  
*1-channel 8-bit unsigned char image min with its X and Y coordinates.*
- **NppStatus nppiMinIdx\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** \*pMin, int \*pIndexX, int \*pIndexY)  
*1-channel 16-bit unsigned short integer image min with its X and Y coordinates.*
- **NppStatus nppiMinIdx\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** \*pMin, int \*pIndexX, int \*pIndexY)  
*1-channel 16-bit signed short integer image min with its X and Y coordinates.*
- **NppStatus nppiMinIdx\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** \*pMin, int \*pIndexX, int \*pIndexY)  
*1-channel 32-bit floating point image min with its X and Y coordinates.*
- **NppStatus nppiMinIdx\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])  
*3-channel 8-bit unsigned char image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])  
*3-channel 16-bit unsigned short integer image min values with their X and Y coordinates.*

- **NppStatus nppiMinIdx\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])  
*3-channel 16-bit signed short integer image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])  
*3-channel 32-bit floating point image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[4], int aIndexX[4], int aIndexY[4])  
*4-channel 8-bit unsigned char image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[4], int aIndexX[4], int aIndexY[4])  
*4-channel 16-bit unsigned short integer image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[4], int aIndexX[4], int aIndexY[4])  
*4-channel 16-bit signed short integer image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[4], int aIndexX[4], int aIndexY[4])  
*4-channel 32-bit floating point image min values with their X and Y coordinates.*
- **NppStatus nppiMinIdx\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])  
*4-channel 8-bit unsigned char image min values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMinIdx\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])  
*4-channel 16-bit unsigned short integer image min values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMinIdx\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])  
*4-channel 16-bit signed short integer image min values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMinIdx\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])  
*4-channel 32-bit floating point image min values with their X and Y coordinates (alpha channel is not processed).*

## 7.77.1 Function Documentation

### 7.77.1.1 **NppStatus nppiMin\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3])

4-channel 16-bit signed short integer image min (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.77.1.2 NppStatus nppiMin\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s \* pMin)

1-channel 16-bit signed short integer image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.77.1.3 NppStatus nppiMin\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMin[3])

3-channel 16-bit signed short integer image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.77.1.4 `NppStatus nppiMin_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[4])`

4-channel 16-bit signed short integer image min.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.77.1.5 `NppStatus nppiMin_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3])`

4-channel 16-bit unsigned short integer image min (alpha channel is not processed).

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.77.1.6 `NppStatus nppiMin_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMin)`

1-channel 16-bit unsigned short integer image min.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.7 NppStatus nppiMin\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[3])**

3-channel 16-bit unsigned short integer image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.8 NppStatus nppiMin\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[4])**

4-channel 16-bit unsigned short integer image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.9 NppStatus nppiMin\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[3])**

4-channel 32-bit floating point image min (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.10 NppStatus nppiMin\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f \* pMin)**

1-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.11 NppStatus nppiMin\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[3])**

3-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.12 NppStatus nppiMin\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[4])**

4-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.13 NppStatus nppiMin\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMin[3])**

4-channel 8-bit unsigned char image min (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_8u\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.14 NppStatus nppiMin\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u \* pMin)**

1-channel 8-bit unsigned char image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_8u\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.15 NppStatus nppiMin\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMin[3])**

3-channel 8-bit unsigned char image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_8u\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.16** `NppStatus nppiMin_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4])`

4-channel 8-bit unsigned char image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_8u\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.17** `NppStatus nppiMinGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiMin\_16s\_AC4R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.18** `NppStatus nppiMinGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiMin\_16s\_C1R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.19 NppStatus nppiMinGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.20 NppStatus nppiMinGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.21 NppStatus nppiMinGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.22 NppStatus nppiMinGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.23 NppStatus nppiMinGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMin\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.24 NppStatus nppiMinGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMin\_16u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.25 NppStatus nppiMinGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMin\_32f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.26 NppStatus nppiMinGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.27 NppStatus nppiMinGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.28 NppStatus nppiMinGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.29 NppStatus nppiMinGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMin\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.30 NppStatus nppiMinGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMin\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.31 NppStatus nppiMinGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMin\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.32 NppStatus nppiMinGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMin\_8u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.33** `NppStatus nppiMinIndx_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int aIndexY[3])`

4-channel 16-bit signed short integer image min values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for the three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for the three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.34** `NppStatus nppiMinIndx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMin, int * pIndexX, int * pIndexY)`

1-channel 16-bit signed short integer image min with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16s\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

*pIndexX* Device-memory pointer to the X coordinate of the image min value.

*pIndexY* Device-memory pointer to the Y coordinate of the image min value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.35** `NppStatus nppiMinIndx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int aIndexY[3])`

3-channel 16-bit signed short integer image min values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16s\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.36** `NppStatus nppiMinIdx_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMin[4], int aIndexX[4], int aIndexY[4])`

4-channel 16-bit signed short integer image min values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16s\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.37** `NppStatus nppiMinIdx_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])`

4-channel 16-bit unsigned short integer image min values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for the three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for the three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.38** `NppStatus nppiMinIdx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMin, int * pIndexX, int * pIndexY)`

1-channel 16-bit unsigned short integer image min with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16u\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

*pIndexX* Device-memory pointer to the X coordinate of the image min value.

*pIndexY* Device-memory pointer to the Y coordinate of the image min value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.39** `NppStatus nppiMinIdx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])`

3-channel 16-bit unsigned short integer image min values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16u\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.40 NppStatus nppiMinIndx\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[4], int aIndexX[4], int aIndexY[4])**

4-channel 16-bit unsigned short integer image min values with their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16u\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.41 NppStatus nppiMinIndx\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[3], int aIndexX[3], int aIndexY[3])**

4-channel 32-bit floating point image min values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for the three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for the three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.42** `NppStatus nppiMinIndx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMin, int * pIndexX, int * pIndexY)`

1-channel 32-bit floating point image min with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

*pIndexX* Device-memory pointer to the X coordinate of the image min value.

*pIndexY* Device-memory pointer to the Y coordinate of the image min value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.43** `NppStatus nppiMinIndx_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3], int aIndexX[3], int aIndexY[3])`

3-channel 32-bit floating point image min values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.44** `NppStatus nppiMinIndx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4], int aIndexX[4], int aIndexY[4])`

4-channel 32-bit floating point image min values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.45** `NppStatus nppiMinIdx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])`

4-channel 8-bit unsigned char image min values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for the three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for the three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for the three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.46** `NppStatus nppiMinIdx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMin, int * pIndexX, int * pIndexY)`

1-channel 8-bit unsigned char image min with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pMin* Device-memory pointer receiving the minimum result.

*pIndexX* Device-memory pointer to the X coordinate of the image min value.

*pIndexY* Device-memory pointer to the Y coordinate of the image min value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.47** `NppStatus nppiMinIdx_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])`

3-channel 8-bit unsigned char image min values with their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_C3R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.48** `NppStatus nppiMinIdx_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4], int aIndexX[4], int aIndexY[4])`

4-channel 8-bit unsigned char image min values with their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image min values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image min values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.77.1.49 NppStatus nppiMinIndxGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinIndx\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.50 NppStatus nppiMinIndxGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.51 NppStatus nppiMinIndxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.52 NppStatus nppiMinIndxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.53 NppStatus nppiMinIdxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinIdx\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.54 NppStatus nppiMinIdxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinIdx\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.55 NppStatus nppiMinIdxGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinIdx\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.56 NppStatus nppiMinIdxGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIdx\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.57 NppStatus nppiMinIdxGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.58 NppStatus nppiMinIdxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.59 NppStatus nppiMinIdxGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.60 NppStatus nppiMinIndxGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinIndx\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.61 NppStatus nppiMinIndxGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinIndx\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.62 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinIndx\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.63 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.77.1.64 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.78 Maximum

### Max

These max routines find the maximal pixel value of an image.

If the image has multiple channels, the functions find the maximum for each channel separately. The scratch buffer is also required by the functions.

- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_8u\_C1R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16u\_C1R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16s\_C1R.*
- [NppStatus nppiMaxGetBufferHostSize\\_32f\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_32f\_C1R.*
- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_8u\_C3R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16u\_C3R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16s\_C3R.*
- [NppStatus nppiMaxGetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_32f\_C3R.*
- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_8u\_C4R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16u\_C4R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16s\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16s\_C4R.*
- [NppStatus nppiMaxGetBufferHostSize\\_32f\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_32f\_C4R.*
- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_8u\_AC4R.*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16u\_AC4R.*

- **NppStatus nppiMaxGetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_16s\_AC4R.*
- **NppStatus nppiMaxGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMax\_32f\_AC4R.*
- **NppStatus nppiMax\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** \*pMax)  
*1-channel 8-bit unsigned char image max.*
- **NppStatus nppiMax\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** \*pMax)  
*1-channel 16-bit unsigned short integer image max.*
- **NppStatus nppiMax\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** \*pMax)  
*1-channel 16-bit signed short integer image max.*
- **NppStatus nppiMax\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** \*pMax)  
*1-channel 32-bit floating point image max.*
- **NppStatus nppiMax\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[3])  
*3-channel 8-bit unsigned char image max.*
- **NppStatus nppiMax\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[3])  
*3-channel 16-bit unsigned short integer image max.*
- **NppStatus nppiMax\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3])  
*3-channel 16-bit signed short integer image max.*
- **NppStatus nppiMax\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[3])  
*3-channel 32-bit floating point image max.*
- **NppStatus nppiMax\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[4])  
*4-channel 8-bit unsigned char image max.*
- **NppStatus nppiMax\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[4])  
*4-channel 16-bit unsigned short integer image max.*
- **NppStatus nppiMax\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[4])  
*4-channel 16-bit signed short integer image max.*

- `NppStatus nppiMax_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMax[4]`)  
*4-channel 32-bit floating point image max.*
- `NppStatus nppiMax_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp8u aMax[3]`)  
*4-channel 8-bit unsigned char image max (alpha channel is not processed).*
- `NppStatus nppiMax_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16u aMax[3]`)  
*4-channel 16-bit unsigned short integer image max (alpha channel is not processed).*
- `NppStatus nppiMax_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp16s aMax[3]`)  
*4-channel 16-bit signed short integer image max (alpha channel is not processed).*
- `NppStatus nppiMax_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp32f aMax[3]`)  
*4-channel 32-bit floating point image max (alpha channel is not processed).*

## MaxIndx

The functions find the max value and its indices (X and Y coordinates) of an image.

If the image contains multiple channels, the functions finds the values and their indices for each channel separately. If there are several maxima in the selected region of interest, the function returns the top leftmost position.

- `NppStatus nppiMaxIndxGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C1R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C1R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C1R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C1R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C3R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C3R.*
- `NppStatus nppiMaxIndxGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, int `*hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C3R.*

- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C3R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_AC4R.*
- [NppStatus nppiMaxIndx\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp8u](#) \*pMax, int \*pIndexX, int \*pIndexY)  
*1-channel 8-bit unsigned char image max value with its X and Y coordinates.*
- [NppStatus nppiMaxIndx\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16u](#) \*pMax, int \*pIndexX, int \*pIndexY)  
*1-channel 16-bit unsigned short integer image max value with its X and Y coordinates.*
- [NppStatus nppiMaxIndx\\_16s\\_C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16s](#) \*pMax, int \*pIndexX, int \*pIndexY)  
*1-channel 16-bit signed short integer image max value with its X and Y coordinates.*
- [NppStatus nppiMaxIndx\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp32f](#) \*pMax, int \*pIndexX, int \*pIndexY)  
*1-channel 32-bit floating point image max value with its X and Y coordinates.*
- [NppStatus nppiMaxIndx\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp8u](#) aMax[3], int aIndexX[3], int aIndexY[3])  
*3-channel 8-bit unsigned char image max values with their X and Y coordinates.*
- [NppStatus nppiMaxIndx\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16u](#) aMax[3], int aIndexX[3], int aIndexY[3])  
*3-channel 16-bit unsigned short integer image max values with their X and Y coordinates.*

- **NppStatus nppiMaxIndx\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])  
*3-channel 16-bit signed short integer image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])  
*3-channel 32-bit floating point image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[4], int aIndexX[4], int aIndexY[4])  
*4-channel 8-bit unsigned char image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[4], int aIndexX[4], int aIndexY[4])  
*4-channel 16-bit unsigned short integer image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[4], int aIndexX[4], int aIndexY[4])  
*4-channel 16-bit signed short integer image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[4], int aIndexX[4], int aIndexY[4])  
*4-channel 32-bit floating point image max values with their X and Y coordinates.*
- **NppStatus nppiMaxIndx\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])  
*4-channel 8-bit unsigned char image max values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMaxIndx\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])  
*4-channel 16-bit unsigned short integer image max values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMaxIndx\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])  
*4-channel 16-bit signed short integer image max values with their X and Y coordinates (alpha channel is not processed).*
- **NppStatus nppiMaxIndx\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])  
*4-channel 32-bit floating point image max values with their X and Y coordinates (alpha channel is not processed).*

### 7.78.1 Function Documentation

#### 7.78.1.1 **NppStatus nppiMax\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3])

4-channel 16-bit signed short integer image max (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.78.1.2 NppStatus nppiMax\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s \* pMax)

1-channel 16-bit signed short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C1R](#) to determine the minimum number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.78.1.3 NppStatus nppiMax\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMax[3])

3-channel 16-bit signed short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C3R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.4 NppStatus nppiMax\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMax[4])**

4-channel 16-bit signed short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.5 NppStatus nppiMax\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMax[3])**

4-channel 16-bit unsigned short integer image max (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.6 NppStatus nppiMax\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u \* pMax)**

1-channel 16-bit unsigned short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.7 NppStatus nppiMax\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMax[3])**

3-channel 16-bit unsigned short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.8 NppStatus nppiMax\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMax[4])**

4-channel 16-bit unsigned short integer image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.9 NppStatus nppiMax\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMax[3])**

4-channel 32-bit floating point image max (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.10** `NppStatus nppiMax_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMax)`

1-channel 32-bit floating point image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.11** `NppStatus nppiMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])`

3-channel 32-bit floating point image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.12** `NppStatus nppiMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4])`

4-channel 32-bit floating point image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.13 NppStatus nppiMax\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMax[3])**

4-channel 8-bit unsigned char image max (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_AC4R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.14 NppStatus nppiMax\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u \* pMax)**

1-channel 8-bit unsigned char image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_C1R](#) to determine the minimum number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.15 NppStatus nppiMax\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMax[3])**

3-channel 8-bit unsigned char image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_C3R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum results, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.16** `NppStatus nppiMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4])`

4-channel 8-bit unsigned char image max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMaxGetBufferHostSize_8u_C4R` to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum results, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.17** `NppStatus nppiMaxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiMax\_16s\_AC4R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.18** `NppStatus nppiMaxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiMax\_16s\_C1R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.19 NppStatus nppiMaxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.20 NppStatus nppiMaxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.21 NppStatus nppiMaxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.22 NppStatus nppiMaxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.23 NppStatus nppiMaxGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMax\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.24 NppStatus nppiMaxGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMax\_16u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.25 NppStatus nppiMaxGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMax\_32f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.26 NppStatus nppiMaxGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.27 NppStatus nppiMaxGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.28 NppStatus nppiMaxGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.29 NppStatus nppiMaxGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMax\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.30 NppStatus nppiMaxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMax\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.31 NppStatus nppiMaxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMax\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.32 NppStatus nppiMaxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMax\_8u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.33** `NppStatus nppiMaxIdx_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3], int aIndexX[3], int aIndexY[3])`

4-channel 16-bit signed short integer image max values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_AC4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for the first three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.34** `NppStatus nppiMaxIdx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMax, int * pIndexX, int * pIndexY)`

1-channel 16-bit signed short integer image max value with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

*pIndexX* Device-memory pointer to the X coordinate of the image max value.

*pIndexY* Device-memory pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.35** `NppStatus nppiMaxIdx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3], int aIndexX[3], int aIndexY[3])`

3-channel 16-bit signed short integer image max values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_C3R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.36** `NppStatus nppiMaxIdx_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[4], int aIndexX[4], int aIndexY[4])`

4-channel 16-bit signed short integer image max values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_C4R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.37** `NppStatus nppiMaxIdx_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])`

4-channel 16-bit unsigned short integer image max values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for the first three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.38** `NppStatus nppiMaxIndx_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u * pMax, int * pIndexX, int * pIndexY)`

1-channel 16-bit unsigned short integer image max value with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

*pIndexX* Device-memory pointer to the X coordinate of the image max value.

*pIndexY* Device-memory pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.39** `NppStatus nppiMaxIndx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])`

3-channel 16-bit unsigned short integer image max values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.40** `NppStatus nppiMaxIdx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4], int aIndexX[4], int aIndexY[4])`

4-channel 16-bit unsigned short integer image max values with their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.41** `NppStatus nppiMaxIdx_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3], int aIndexX[3], int aIndexY[3])`

4-channel 32-bit floating point image max values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for the first three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.42** `NppStatus nppiMaxIdx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f * pMax, int * pIndexX, int * pIndexY)`

1-channel 32-bit floating point image max value with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.

*pIndexX* Device-memory pointer to the X coordinate of the image max value.

*pIndexY* Device-memory pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.43** `NppStatus nppiMaxIdx_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3], int aIndexX[3], int aIndexY[3])`

3-channel 32-bit floating point image max values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.44** `NppStatus nppiMaxIdx_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4], int aIndexX[4], int aIndexY[4])`

4-channel 32-bit floating point image max values with their X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_C4R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, four elements for four channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.45** `NppStatus nppiMaxIdx_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])`

4-channel 8-bit unsigned char image max values with their X and Y coordinates (alpha channel is not processed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_AC4R](#) to determine the minimum number of bytes required.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for the first three channels.

*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.46** `NppStatus nppiMaxIdx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMax, int * pIndexX, int * pIndexY)`

1-channel 8-bit unsigned char image max value with its X and Y coordinates.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_C1R](#) to determine the minimum number of bytes required.

*pMax* Device-memory pointer receiving the maximum result.  
*pIndexX* Device-memory pointer to the X coordinate of the image max value.  
*pIndexY* Device-memory pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.47** `NppStatus nppiMaxIdx_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])`

3-channel 8-bit unsigned char image max values with their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.  
*aMax* Device-memory array receiving the maximum result, three elements for three channels.  
*aIndexX* Device-memory array to the X coordinates of the image max values, three elements for three channels.  
*aIndexY* Device-memory array to the Y coordinates of the image max values, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.48** `NppStatus nppiMaxIdx_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4], int aIndexX[4], int aIndexY[4])`

4-channel 8-bit unsigned char image max valueswith their X and Y coordinates.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_C4R](#) to determine the minium number of bytes required.  
*aMax* Device-memory array receiving the maximum result, four elements for four channels.  
*aIndexX* Device-memory array to the X coordinates of the image max values, four elements for four channels.  
*aIndexY* Device-memory array to the Y coordinates of the image max values, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.78.1.49 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.50 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.51 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.52 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.53** NppStatus nppiMaxIndxGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.54** NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.55** NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.56 NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.57 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.58 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.59 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.60** NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.61** NppStatus nppiMaxIndxGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.62** NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.63 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.78.1.64 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.79 Minimum\_Maximum

### MinMax

The functions find the minimum and maximum values of an image.

If the image contains multiple channels, the function find the values for each channel separately. The functions also require the device scratch buffer.

- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_8u\_C1R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16u\_C1R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16s\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C1R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_32f\_C1R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_8u\_C3R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16u\_C3R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16s\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C3R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_32f\_C3R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_8u\_AC4R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16u\_AC4R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16s\_AC4R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_32f\_AC4R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_8u\_C4R.*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16u\_C4R.*

- **NppStatus nppiMinMaxGetBufferHostSize\_16s\_C4R** (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C4R.*
- **NppStatus nppiMinMaxGetBufferHostSize\_32f\_C4R** (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinManx\_32f\_C4R.*
- **NppStatus nppiMinMax\_8u\_C1R** (const Npp8u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \*pMin, Npp8u \*pMax, Npp8u \*pDeviceBuffer)  
*1-channel 8-bit unsigned image minimum and maximum values.*
- **NppStatus nppiMinMax\_16u\_C1R** (const Npp16u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u \*pMin, Npp16u \*pMax, Npp8u \*pDeviceBuffer)  
*1-channel 16-bit unsigned short image minimum and maximum values.*
- **NppStatus nppiMinMax\_16s\_C1R** (const Npp16s \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s \*pMin, Npp16s \*pMax, Npp8u \*pDeviceBuffer)  
*1-channel 16-bit signed short image minimum and maximum values.*
- **NppStatus nppiMinMax\_32f\_C1R** (const Npp32f \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f \*pMin, Npp32f \*pMax, Npp8u \*pDeviceBuffer)  
*1-channel 32-bit floating point image minimum and maximum values.*
- **NppStatus nppiMinMax\_8u\_C3R** (const Npp8u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u \*pDeviceBuffer)  
*3-channel 8-bit unsigned image minimum and maximum values.*
- **NppStatus nppiMinMax\_16u\_C3R** (const Npp16u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u \*pDeviceBuffer)  
*3-channel 16-bit unsigned short image minimum and maximum values.*
- **NppStatus nppiMinMax\_16s\_C3R** (const Npp16s \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[3], Npp16s aMax[3], Npp8u \*pDeviceBuffer)  
*3-channel 16-bit signed short image minimum and maximum values.*
- **NppStatus nppiMinMax\_32f\_C3R** (const Npp32f \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u \*pDeviceBuffer)  
*3-channel 32-bit floating point image minimum and maximum values.*
- **NppStatus nppiMinMax\_8u\_AC4R** (const Npp8u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u \*pDeviceBuffer)  
*4-channel 8-bit unsigned image minimum and maximum values (alpha channel is not calculated).*
- **NppStatus nppiMinMax\_16u\_AC4R** (const Npp16u \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u \*pDeviceBuffer)  
*4-channel 16-bit unsigned short image minimum and maximum values (alpha channel is not calculated).*
- **NppStatus nppiMinMax\_16s\_AC4R** (const Npp16s \*pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[3], Npp16s aMax[3], Npp8u \*pDeviceBuffer)  
*4-channel 16-bit signed short image minimum and maximum values (alpha channel is not calculated).*

- `NppStatus nppiMinMax_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[3], `Npp32f` aMax[3], `Npp8u` \*pDeviceBuffer)  
*4-channel 32-bit floating point image minimum and maximum values (alpha channel is not calculated).*
- `NppStatus nppiMinMax_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[4], `Npp8u` aMax[4], `Npp8u` \*pDeviceBuffer)  
*4-channel 8-bit unsigned image minimum and maximum values.*
- `NppStatus nppiMinMax_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[4], `Npp16u` aMax[4], `Npp8u` \*pDeviceBuffer)  
*4-channel 16-bit unsigned short image minimum and maximum values.*
- `NppStatus nppiMinMax_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[4], `Npp16s` aMax[4], `Npp8u` \*pDeviceBuffer)  
*4-channel 16-bit signed short image minimum and maximum values.*
- `NppStatus nppiMinMax_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[4], `Npp32f` aMax[4], `Npp8u` \*pDeviceBuffer)  
*4-channel 32-bit floating point image minimum and maximum values.*

## MinMaxIndx

MinMax value and their indices (X and Y coordinates) of images.

If there are several minima and maxima in the selected region of interest, the function returns the top leftmost position.

- `NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_8u\_C1R.*
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_8s\_C1R.*
- `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_16u\_C1R.*
- `NppStatus nppiMinMaxIndxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_32f\_C1R.*
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_8u\_C1MR.*
- `NppStatus nppiMinMaxIndxGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIndx\_8s\_C1MR.*

- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C1MR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C1MR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_32f\\_C1MR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_32f\_C1MR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8u\_C3CR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8s\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8s\_C3CR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C3CR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_32f\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_32f\_C3CR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8u\_C3CMR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8s\_C3CMR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C3CMR.*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMinMaxIdx\_32f\_C3CMR.*
- **NppStatus** [nppiMinMaxIdx\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pMinValue, [Npp8u](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)  
*1-channel 8-bit unsigned char image minimum and maximum values with their indices.*
- **NppStatus** [nppiMinMaxIdx\\_8s\\_C1R](#) (const [Npp8s](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8s](#) \*pMinValue, [Npp8s](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)  
*1-channel 8-bit signed char image minimum and maximum values with their indices.*

- `NppStatus nppiMinMaxIndx_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` \*pMinValue, `Npp16u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 16-bit unsigned short image minimum and maximum values with their indices.*
- `NppStatus nppiMinMaxIndx_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` \*pMinValue, `Npp32f` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating point image minimum and maximum values with their indices.*
- `NppStatus nppiMinMaxIndx_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pMinValue, `Npp8u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 8-bit unsigned char image minimum and maximum values with their indices, [Masked Operation](#).*
- `NppStatus nppiMinMaxIndx_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8s` \*pMinValue, `Npp8s` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 8-bit signed char image minimum and maximum values with their indices, [Masked Operation](#).*
- `NppStatus nppiMinMaxIndx_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp16u` \*pMinValue, `Npp16u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 16-bit unsigned short image minimum and maximum values with their indices, [Masked Operation](#).*
- `NppStatus nppiMinMaxIndx_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp32f` \*pMinValue, `Npp32f` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating point image minimum and maximum values with their indices, [Masked Operation](#).*
- `NppStatus nppiMinMaxIndx_8u_C3CR` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pMinValue, `Npp8u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit unsigned char image minimum and maximum values with their indices, [Channel-of-Interest API](#).*
- `NppStatus nppiMinMaxIndx_8s_C3CR` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8s` \*pMinValue, `Npp8s` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit signed char image minimum and maximum values with their indices, [Channel-of-Interest API](#).*
- `NppStatus nppiMinMaxIndx_16u_C3CR` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp16u` \*pMinValue, `Npp16u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 16-bit unsigned short image minimum and maximum values with their indices, [Channel-of-Interest API](#).*
- `NppStatus nppiMinMaxIndx_32f_C3CR` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp32f` \*pMinValue, `Npp32f` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating point image minimum and maximum values with their indices, Channel-of-Interest API.*

- `NppStatus nppiMinMaxIndx_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pMinValue, `Npp8u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit unsigned char image minimum and maximum values with their indices, Masked Operation, Channel-of-Interest API.*

- `NppStatus nppiMinMaxIndx_8s_C3CMR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8s` \*pMinValue, `Npp8s` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit signed char image minimum and maximum values with their indices, Masked Operation, Channel-of-Interest API.*

- `NppStatus nppiMinMaxIndx_16u_C3CMR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp16u` \*pMinValue, `Npp16u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit unsigned short image minimum and maximum values with their indices, Masked Operation, Channel-of-Interest API.*

- `NppStatus nppiMinMaxIndx_32f_C3CMR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp32f` \*pMinValue, `Npp32f` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*3-channel 32-bit floating point image minimum and maximum values with their indices, Masked Operation, Channel-of-Interest API.*

## 7.79.1 Function Documentation

### 7.79.1.1 `NppStatus nppiMinMax_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[3], `Npp16s` aMax[3], `Npp8u` \*pDeviceBuffer)

4-channel 16-bit signed short image minimum and maximum values (alpha channel is not calculated).

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use `nppiMinMaxGetBufferHostSize_16s_AC4R` to determine the minimum number of bytes required.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.79.1.2 NppStatus nppiMinMax\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s \* pMin, Npp16s \* pMax, Npp8u \* pDeviceBuffer)**

1-channel 16-bit signed short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMin* Device-memory pointer receiving the minimum result.

*pMax* Device-memory pointer receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.3 NppStatus nppiMinMax\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[3], Npp16s aMax[3], Npp8u \* pDeviceBuffer)**

3-channel 16-bit signed short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.4 NppStatus nppiMinMax\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16s aMin[4], Npp16s aMax[4], Npp8u \* pDeviceBuffer)**

4-channel 16-bit signed short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C4R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.5 NppStatus nppiMinMax\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u \* pDeviceBuffer)**

4-channel 16-bit unsigned short image minimum and maximum values (alpha channel is not calculated).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.6 NppStatus nppiMinMax\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u \* pMin, Npp16u \* pMax, Npp8u \* pDeviceBuffer)**

1-channel 16-bit unsigned short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMin* Device-memory pointer receiving the minimum result.

*pMax* Device-memory pointer receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.7 NppStatus nppiMinMax\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u \* pDeviceBuffer)**

3-channel 16-bit unsigned short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.8 NppStatus nppiMinMax\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[4], Npp16u aMax[4], Npp8u \* pDeviceBuffer)**

4-channel 16-bit unsigned short image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.9 NppStatus nppiMinMax\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u \* pDeviceBuffer)**

4-channel 32-bit floating point image minimum and maximum values (alpha channel is not calculated).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.10 NppStatus nppiMinMax\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f \* pMin, Npp32f \* pMax, Npp8u \* pDeviceBuffer)**

1-channel 32-bit floating point image minimum and maximum values.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMin* Device-memory pointer receiving the minimum result.

*pMax* Device-memory pointer receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.11 NppStatus nppiMinMax\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u \* pDeviceBuffer)**

3-channel 32-bit floating point image minimum and maximum values.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Device-memory array receiving the minimum result, three elements for three channels..

*aMax* Device-memory array receiving the maximum result, three elements for three channels..

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_C3R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.12** `NppStatus nppiMinMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[4], Npp32f aMax[4], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating point image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*pDeviceBuffer* Buffer to a scratch memory. Use `nppiMinMaxGetBufferHostSize_32f_C4R` to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.13** `NppStatus nppiMinMax_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned image minimum and maximum values (alpha channel is not calculated).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Device-memory array receiving the minimum result, three elements for the first three channels.

*aMax* Device-memory array receiving the maximum result, three elements for the first three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use `nppiMinMaxGetBufferHostSize_8u_AC4R` to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.14** `NppStatus nppiMinMax_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMin, Npp8u * pMax, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned image minimum and maximum values.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMin* Device-memory pointer receiving the minimum result.

*pMax* Device-memory pointer receiving the maximum result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.15** `NppStatus nppiMinMax_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned image minimum and maximum values.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Device-memory array receiving the minimum result, three elements for three channels.

*aMax* Device-memory array receiving the maximum result, three elements for three channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C3R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.16** `NppStatus nppiMinMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[4], Npp8u aMax[4], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned image minimum and maximum values.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Device-memory array receiving the minimum result, four elements for four channels.

*aMax* Device-memory array receiving the maximum result, four elements for four channels.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C4R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.79.1.17 NppStatus nppiMinMaxGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinManx\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.18 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.19 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.20 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinManx\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.21 NppStatus nppiMinMaxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinManx\_16u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.22 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinManx\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.23 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiMinManx\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.24 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinManx\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.25 NppStatus nppiMinMaxGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinManx\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.26 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinManx\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.27 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinManx\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.28 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinManx\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.29 NppStatus nppiMinMaxGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinManx\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.30 NppStatus nppiMinMaxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinManx\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.31** `NppStatus nppiMinMaxGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMinManx_8u_C3R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.32** `NppStatus nppiMinMaxGetBufferHostSize_8u_C4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMinManx_8u_C4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.33** `NppStatus nppiMinMaxIndx_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 16-bit unsigned short image minimum and maximum values with their indices, [Masked Operation](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e.,  $pMinIndex = \{0, 0\}$ ,  $pMaxIndex = \{0, 0\}$ ,  $pMinValue = 0$ ,  $pMaxValue = 0$ . If any of  $pMinValue$ ,  $pMaxValue$ ,  $pMinIndex$ , or  $pMaxIndex$  is not needed, zero pointer must be passed correspondingly.

**7.79.1.34 NppStatus nppiMinMaxIndx\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u \* pMinValue, Npp16u \* pMaxValue, NppiPoint \* pMinIndex, NppiPoint \* pMaxIndex, Npp8u \* pDeviceBuffer)**

1-channel 16-bit unsigned short image minimum and maximum values with their indices.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of  $pMinValue$ ,  $pMaxValue$ ,  $pMinIndex$ , or  $pMaxIndex$  is not needed, zero pointer must be passed correspondingly.

**7.79.1.35 NppStatus nppiMinMaxIndx\_16u\_C3CMR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp16u \* pMinValue, Npp16u \* pMaxValue, NppiPoint \* pMinIndex, NppiPoint \* pMaxIndex, Npp8u \* pDeviceBuffer)**

3-channel 16-bit unsigned short image minimum and maximum values with their indices, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.36** `NppStatus nppiMinMaxIndx_16u_C3CR (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp16u * pMinValue, Npp16u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 16-bit unsigned short image minimum and maximum values with their indices, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C3CR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.37** `NppStatus nppiMinMaxIndx_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating point image minimum and maximum values with their indices, [Masked Operation](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C1MR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.38** `NppStatus nppiMinMaxIndx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating point image minimum and maximum values with their indices.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.39** `NppStatus nppiMinMaxIndx_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

3-channel 32-bit floating point image minimum and maximum values with their indices, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use `nppiMinMaxIndxGetBufferHostSize_32f_C3CMR` to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.40** `NppStatus nppiMinMaxIndx_32f_C3CR (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating point image minimum and maximum values with their indices, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C3CR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.79.1.41** `NppStatus nppiMinMaxIndx_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 8-bit signed char image minimum and maximum values with their indices, [Masked Operation](#).

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C1MR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.79.1.42** `NppStatus nppiMinMaxIndx_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 8-bit signed char image minimum and maximum values with their indices.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.79.1.43** `NppStatus nppiMinMaxIndx_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

3-channel 8-bit signed char image minimum and maximum values with their indices, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.44** `NppStatus nppiMinMaxIndx_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

3-channel 8-bit signed char image minimum and maximum values with their indices, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C3CR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.45** `NppStatus nppiMinMaxIndx_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image minimum and maximum values with their indices, [Masked Operation](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C1MR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.79.1.46** `NppStatus nppiMinMaxIdx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image minimum and maximum values with their indices.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Device-memory pointer receiving the minimum value.

*pMaxValue* Device-memory pointer receiving the maximum value.

*pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.79.1.47** `NppStatus nppiMinMaxIdx_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned char image minimum and maximum values with their indices, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel\_of\_Interest Number.
- pMinValue* Device-memory pointer receiving the minimum value.
- pMaxValue* Device-memory pointer receiving the maximum value.
- pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.
- pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.
- pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C3CMR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

#### 7.79.1.48 `NppStatus nppiMinMaxIdx_8u_C3CR (const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u *pMinValue, Npp8u *pMaxValue, NppiPoint *pMinIndex, NppiPoint *pMaxIndex, Npp8u *pDeviceBuffer)`

3-channel 8-bit unsigned char image minimum and maximum values with their indices, [Channel-of-Interest API](#).

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel\_of\_Interest Number.
- pMinValue* Device-memory pointer receiving the minimum value.
- pMaxValue* Device-memory pointer receiving the maximum value.
- pMinIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the minimum value.
- pMaxIndex* Device-memory pointer receiving the indices (X and Y coordinates) of the maximum value.
- pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C3CR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.79.1.49 NppStatus nppiMinMaxIdxGetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.50 NppStatus nppiMinMaxIdxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.51 NppStatus nppiMinMaxIdxGetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.52 NppStatus nppiMinMaxIdxGetBufferHostSize\_16u\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_16u\_C3CR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.53 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_32f\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.54 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_32f\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.55 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.56 NppStatus nppiMinMaxIdxGetBufferHostSize\_32f\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_32f\_C3CR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.57 NppStatus nppiMinMaxIdxGetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.58 NppStatus nppiMinMaxIdxGetBufferHostSize\_8s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.59 NppStatus nppiMinMaxIdxGetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.60 NppStatus nppiMinMaxIndxGetBufferHostSize\_8s\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_8s\_C3CR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.61 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_8u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.62 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMinMaxIndx\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.63 NppStatus nppiMinMaxIdxGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.79.1.64 NppStatus nppiMinMaxIdxGetBufferHostSize\_8u\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMinMaxIdx\_8u\_C3CR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.80 Mean

### Mean

The functions compute the mean value of all the pixel values in an image.

All the mean results are stored in a 64-bit double precision format. If the image contains multiple channels, the functions calculate the mean for each channel separately. The mean functions require additional scratch buffer for computations.

- [NppStatus nppiMeanGetBufferHostSize\\_8u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_C1R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16u\_C1R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16s\_C1R.*
- [NppStatus nppiMeanGetBufferHostSize\\_32f\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_C1R.*
- [NppStatus nppiMeanGetBufferHostSize\\_8u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_C3R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16u\_C3R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16s\_C3R.*
- [NppStatus nppiMeanGetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_C3R.*
- [NppStatus nppiMeanGetBufferHostSize\\_8u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_AC4R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16u\_AC4R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16s\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16s\_AC4R.*
- [NppStatus nppiMeanGetBufferHostSize\\_32f\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_AC4R.*
- [NppStatus nppiMeanGetBufferHostSize\\_8u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_C4R.*
- [NppStatus nppiMeanGetBufferHostSize\\_16u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiMean\_16u\_C4R.*

- **NppStatus nppiMeanGetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16s\_C4R.*
- **NppStatus nppiMeanGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_C4R.*
- **NppStatus nppiMeanGetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_C1MR.*
- **NppStatus nppiMeanGetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8s\_C1MR.*
- **NppStatus nppiMeanGetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16u\_C1MR.*
- **NppStatus nppiMeanGetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_C1MR.*
- **NppStatus nppiMeanGetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8u\_C3CMR.*
- **NppStatus nppiMeanGetBufferHostSize\_8s\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_8s\_C3CMR.*
- **NppStatus nppiMeanGetBufferHostSize\_16u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_16u\_C3CMR.*
- **NppStatus nppiMeanGetBufferHostSize\_32f\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_32f\_C3CMR.*
- **NppStatus nppiMean\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pMean)  
*1-channel 8-bit unsigned char image sum with 64-bit double precision result.*
- **NppStatus nppiMean\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pMean)  
*1-channel 16-bit unsigned short integer image mean with 64-bit double precision result.*
- **NppStatus nppiMean\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pMean)  
*1-channel 16-bit signed short integer image mean with 64-bit double precision result.*
- **NppStatus nppiMean\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** \*pMean)  
*1-channel 32-bit floating point image mean with 64-bit double precision result.*
- **NppStatus nppiMean\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aMean[3])

*3-channel 8-bit unsigned char image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*3-channel 16-bit unsigned short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*3-channel 16-bit signed short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*3-channel 32-bit floating point image mean with 64-bit double precision result.*

- `NppStatus nppiMean_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*4-channel 8-bit unsigned char image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*4-channel 16-bit unsigned short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*4-channel 16-bit signed short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

*4-channel 32-bit floating point image mean with 64-bit double precision result.*

- `NppStatus nppiMean_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[4])

*4-channel 8-bit unsigned char image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[4])

*4-channel 16-bit unsigned short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[4])

*4-channel 16-bit signed short image mean with 64-bit double precision result.*

- `NppStatus nppiMean_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[4])

*4-channel 32-bit floating point image mean with 64-bit double precision result.*

- `NppStatus nppiMean_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)

*1-channel 8-bit unsigned char image mean with 64-bit double precision result, Masked Operation.*

- `NppStatus nppiMean_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*1-channel 8-bit signed char image mean with 64-bit double precision result, Masked Operation.*
- `NppStatus nppiMean_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*1-channel 16-bit unsigned short integer image mean with 64-bit double precision result, Masked Operation.*
- `NppStatus nppiMean_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*1-channel 32-bit floating point image mean with 64-bit double precision result, Masked Operation.*
- `NppStatus nppiMean_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*3-channel 8-bit unsigned char image mean with 64-bit double precision result, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiMean_8s_C3CMR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*3-channel 8-bit signed char image mean with 64-bit double precision result, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiMean_16u_C3CMR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*3-channel 16-bit unsigned short integer image mean with 64-bit double precision result, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiMean_32f_C3CMR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*3-channel 32-bit floating point image mean with 64-bit double precision result, Masked Operation, Channel-of-Interest API.*

## 7.80.1 Function Documentation

### 7.80.1.1 `NppStatus nppiMean_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])

4-channel 16-bit signed short image mean with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_16s_AC4R` to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.2 NppStatus nppiMean\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

1-channel 16-bit signed short integer image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.3 NppStatus nppiMean\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])**

3-channel 16-bit signed short image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for three channels.

**7.80.1.4 NppStatus nppiMean\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[4])**

4-channel 16-bit signed short image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.5 NppStatus nppiMean\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])**

4-channel 16-bit unsigned short image mean with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.6 NppStatus nppiMean\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

1-channel 16-bit unsigned short integer image mean with 64-bit double precision result, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.80.1.7 NppStatus nppiMean\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

1-channel 16-bit unsigned short integer image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C1R](#) to determine the minimum number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.8 NppStatus nppiMean\_16u\_C3CMR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

3-channel 16-bit unsigned short integer image mean with 64-bit double precision result, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minimum number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.80.1.9 NppStatus nppiMean\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])**

3-channel 16-bit unsigned short image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.10 NppStatus nppiMean\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[4])**

4-channel 16-bit unsigned short image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.11 NppStatus nppiMean\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])**

4-channel 32-bit floating point image mean with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [NPP\\_NOT\\_EVEN\\_STEP\\_ERROR](#) if an invalid floating-point image is specified.

**7.80.1.12** `NppStatus nppiMean_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

1-channel 32-bit floating point image mean with 64-bit double precision result, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C1MR](#) to determine the minimum number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.80.1.13** `NppStatus nppiMean_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

1-channel 32-bit floating point image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)

*pMean* Device-memory pointer receiving the mean result. Use [nppiMeanGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.80.1.14** `NppStatus nppiMean_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

3-channel 32-bit floating point image mean with 64-bit double precision result, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C3CMR](#) to determine the minium number of bytes required.  
*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

#### 7.80.1.15 NppStatus nppiMean\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])

3-channel 32-bit floating point image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.  
*aMean* Array that contains computed mean, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

#### 7.80.1.16 NppStatus nppiMean\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[4])

4-channel 32-bit floating point image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.  
*aMean* Array that contains computed mean, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.80.1.17** `NppStatus nppiMean_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

1-channel 8-bit signed char image mean with 64-bit double precision result, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8s\\_C1MR](#) to determine the minimum number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.80.1.18** `NppStatus nppiMean_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

3-channel 8-bit signed char image mean with 64-bit double precision result, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minimum number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.80.1.19** `NppStatus nppiMean_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

4-channel 8-bit unsigned char image mean with 64-bit double precision result.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_8u_AC4R` to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for the first three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.20** `NppStatus nppiMean_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

1-channel 8-bit unsigned char image mean with 64-bit double precision result, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_8u_C1MR` to determine the minium number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.80.1.21** `NppStatus nppiMean_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

1-channel 8-bit unsigned char image sum with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.22** `NppStatus nppiMean_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

3-channel 8-bit unsigned char image mean with 64-bit double precision result, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Device-memory pointer receiving the mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.80.1.23** `NppStatus nppiMean_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

3-channel 8-bit unsigned char image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains computed mean, three elements for three channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.24** `NppStatus nppiMean_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

4-channel 8-bit unsigned char image mean with 64-bit double precision result.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_8u_C4R` to determine the minium number of bytes required.

*aMean* Array that contains computed mean, four elements for four channels.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.80.1.25** `NppStatus nppiMeanGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMean_16s_AC4R`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.26** `NppStatus nppiMeanGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMean_16s_C1R`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.27 NppStatus nppiMeanGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.28 NppStatus nppiMeanGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.29 NppStatus nppiMeanGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.30 NppStatus nppiMeanGetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.31 NppStatus nppiMeanGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.32 NppStatus nppiMeanGetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_16u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.33 NppStatus nppiMeanGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.34 NppStatus nppiMeanGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.35 NppStatus nppiMeanGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.36 NppStatus nppiMeanGetBufferHostSize\_32f\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.37 NppStatus nppiMeanGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.38 NppStatus nppiMeanGetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.39 NppStatus nppiMeanGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_32f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.40 NppStatus nppiMeanGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.41 NppStatus nppiMeanGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.42 NppStatus nppiMeanGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.43 NppStatus nppiMeanGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.44 NppStatus nppiMeanGetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.45 NppStatus nppiMeanGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.46 NppStatus nppiMeanGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.47 NppStatus nppiMeanGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.80.1.48 NppStatus nppiMeanGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiMean\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.81 Mean And Standard Deviation

### Mean and Standard Deviation

The routines compute the mean and standard deviation of image pixel values and store them in a 64-bit double precision format.

The functions require the additional device memroy for the computations.

- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8u\_C1R.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C1R.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C1R.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C1R.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8u\_C1MR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C1MR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C1MR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_32f_C1MR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C1MR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_8u_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8u\_C3CR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_8s_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C3CR.*
- `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C3CR` (`NppiSize` `oSizeROI`, `int *hpBufferSize`)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C3CR.*

- **NppStatus** [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C3CR.*
- **NppStatus** [nppiMeanStdDevGetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8u\_C3CMR.*
- **NppStatus** [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C3CMR.*
- **NppStatus** [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C3CMR.*
- **NppStatus** [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C3CMR.*
- **NppStatus** [nppiMean\\_StdDev\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 8-bit unsigned char image mean and standard deviation.*
- **NppStatus** [nppiMean\\_StdDev\\_8s\\_C1R](#) (const [Npp8s](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 8-bit signed char image mean and standard deviation.*
- **NppStatus** [nppiMean\\_StdDev\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 16-bit unsigned short int image mean and standard deviation.*
- **NppStatus** [nppiMean\\_StdDev\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc, [int](#) nSrcStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 32-bit floating-point image mean and standard deviation.*
- **NppStatus** [nppiMean\\_StdDev\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc, [int](#) nSrcStep, const [Npp8u](#) \*pMask, [int](#) nMaskStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 8-bit unsigned char image mean and standard deviation, Masked Operation.*
- **NppStatus** [nppiMean\\_StdDev\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc, [int](#) nSrcStep, const [Npp8u](#) \*pMask, [int](#) nMaskStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 8-bit signed char image mean and standard deviation, Masked Operation.*
- **NppStatus** [nppiMean\\_StdDev\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc, [int](#) nSrcStep, const [Npp8u](#) \*pMask, [int](#) nMaskStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)  
*1-channel 16-bit unsigned short int image mean and standard deviation, Masked Operation.*
- **NppStatus** [nppiMean\\_StdDev\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc, [int](#) nSrcStep, const [Npp8u](#) \*pMask, [int](#) nMaskStep, [NppiSize](#) [oSizeROI](#), [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev)

1-channel 32-bit floating-point image mean and standard deviation, *Masked Operation*.

- `NppStatus nppiMean_StdDev_8u_C3CR` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 8-bit unsigned char image mean and standard deviation, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_8s_C3CR` (const `Npp8s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 8-bit signed char image mean and standard deviation, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_16u_C3CR` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 16-bit unsigned short int image mean and standard deviation, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_32f_C3CR` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 32-bit floating-point image mean and standard deviation, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 8-bit unsigned char image mean and standard deviation, *Masked Operation*, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_8s_C3CMR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 8-bit signed char image mean and standard deviation, *Masked Operation*, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 16-bit unsigned short int image mean and standard deviation, *Masked Operation*, *Channel-of-Interest API*.

- `NppStatus nppiMean_StdDev_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

3-channel 32-bit floating-point image mean and standard deviation, *Masked Operation*, *Channel-of-Interest API*.

## 7.81.1 Function Documentation

- ### 7.81.1.1 `NppStatus nppiMean_StdDev_16u_C1MR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, `Npp8u *pDeviceBuffer`, `Npp64f *pMean`, `Npp64f *pStdDev`)

1-channel 16-bit unsigned short int image mean and standard deviation, *Masked Operation*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.81.1.2 NppStatus nppiMean\_StdDev\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)

1-channel 16-bit unsigned short int image mean and standard deviation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.81.1.3 NppStatus nppiMean\_StdDev\_16u\_C3CMR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)

3-channel 16-bit unsigned short int image mean and standard deviation, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.81.1.4 NppStatus nppiMean\_StdDev\_16u\_C3CR (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

3-channel 16-bit unsigned short int image mean and standard deviation, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.81.1.5 NppStatus nppiMean\_StdDev\_32f\_C1MR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

1-channel 32-bit floating-point image mean and standard deviation, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C1MR](#) to determine the minimum number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.81.1.6 NppStatus nppiMean\_StdDev\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

1-channel 32-bit floating-point image mean and standard deviation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.81.1.7 NppStatus nppiMean\_StdDev\_32f\_C3CMR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

3-channel 32-bit floating-point image mean and standard deviation, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CMR](#) to determine the minimum number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.81.1.8 NppStatus nppiMean\_StdDev\_32f\_C3CR (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

3-channel 32-bit floating-point image mean and standard deviation, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CR](#) to determine the minimum number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.81.1.9 NppStatus nppiMean\_StdDev\_8s\_C1MR (const Npp8s \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean, Npp64f \* pStdDev)**

1-channel 8-bit signed char image mean and standard deviation, [Masked Operation](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.81.1.10** `NppStatus nppiMean_StdDev_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

1-channel 8-bit signed char image mean and standard deviation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C1R](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.81.1.11** `NppStatus nppiMean_StdDev_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

3-channel 8-bit signed char image mean and standard deviation, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.  
*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

#### 7.81.1.12 `NppStatus nppiMean_StdDev_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

3-channel 8-bit signed char image mean and standard deviation, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
 Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CR](#) to determine the minium number of bytes required.  
*pMean* Contains computed mean.  
*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

#### 7.81.1.13 `NppStatus nppiMean_StdDev_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

1-channel 8-bit unsigned char image mean and standard deviation, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
 Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C1MR](#) to determine the minium number of bytes required.  
*pMean* Contains computed mean.  
*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.81.1.14** `NppStatus nppiMean_StdDev_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

1-channel 8-bit unsigned char image mean and standard deviation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.81.1.15** `NppStatus nppiMean_StdDev_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

3-channel 8-bit unsigned char image mean and standard deviation, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.81.1.16** `NppStatus nppiMean_StdDev_8u_C3CR (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean, Npp64f * pStdDev)`

3-channel 8-bit unsigned char image mean and standard deviation, [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use `nppiMeanStdDevGetBufferHostSize_8u_C3CR` to determine the minimum number of bytes required.

*pMean* Contains computed mean.

*pStdDev* Contains computed standard deviation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.81.1.17** `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMean_StdDev_16u_C1MR`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_SUCCESS`

**7.81.1.18** `NppStatus nppiMeanStdDevGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiMean_StdDev_16u_C1R`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_SUCCESS`

**7.81.1.19 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.20 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_16u\_C3CR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.21 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.22 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.23 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.24 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_32f\_C3CR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.25 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_8s\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.26 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.27 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.28 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8s\_C3CR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.29 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiMean\_StdDev\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.30 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.31 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.81.1.32 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiMean\_StdDev\_8u\_C3CR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.82 Infinity Norm

### Infinity Norm

These functions compute the infinity norm of an image.

The infinity norm is defined as the largest pixel value of the image. If the image contains multiple channels, the functions will compute the norm for each channel separately. The functions require the addition device scratch buffer for the computations.

- [NppStatus nppiNormInfGetBufferHostSize\\_8u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C1R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_16u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C1R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_16s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16s\_C1R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_32s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32s\_C1R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_32f\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C1R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_8u\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C1MR.*
- [NppStatus nppiNormInfGetBufferHostSize\\_8s\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8s\_C1MR.*
- [NppStatus nppiNormInfGetBufferHostSize\\_16u\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C1MR.*
- [NppStatus nppiNormInfGetBufferHostSize\\_32f\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C1MR.*
- [NppStatus nppiNormInfGetBufferHostSize\\_8u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C3R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_16u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C3R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_16s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16s\_C3R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C3R.*
- [NppStatus nppiNormInfGetBufferHostSize\\_8u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_AC4R.*

- **NppStatus nppiNormInfGetBufferHostSize\_16u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_AC4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16s\_AC4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_AC4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16s\_C4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C4R.*
- **NppStatus nppiNormInfGetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C3CMR.*
- **NppStatus nppiNormInfGetBufferHostSize\_8s\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_8s\_C3CMR.*
- **NppStatus nppiNormInfGetBufferHostSize\_16u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C3CMR.*
- **NppStatus nppiNormInfGetBufferHostSize\_32f\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C3CMR.*
- **NppStatus nppiNorm\_Inf\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 8-bit unsigned char image infinity norm.*
- **NppStatus nppiNorm\_Inf\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit unsigned short image infinity norm.*
- **NppStatus nppiNorm\_Inf\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit signed short image infinity norm.*
- **NppStatus nppiNorm\_Inf\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)

*1-channel 32-bit signed int image infinity norm.*

- `NppStatus nppiNorm_Inf_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating-point image infinity norm.*

- `NppStatus nppiNorm_Inf_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 8-bit unsigned char image infinity norm, Masked Operation.*

- `NppStatus nppiNorm_Inf_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 8-bit signed char image infinity norm, Masked Operation.*

- `NppStatus nppiNorm_Inf_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 16-bit unsigned short image infinity norm, Masked Operation.*

- `NppStatus nppiNorm_Inf_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating-point image infinity norm, Masked Operation.*

- `NppStatus nppiNorm_Inf_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit unsigned char image infinity norm.*

- `NppStatus nppiNorm_Inf_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit unsigned short image infinity norm.*

- `NppStatus nppiNorm_Inf_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit signed short image infinity norm.*

- `NppStatus nppiNorm_Inf_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 32-bit floating-point image infinity norm.*

- `NppStatus nppiNorm_Inf_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 8-bit unsigned char image infinity norm (alpha channel is not computed).*

- `NppStatus nppiNorm_Inf_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit unsigned short image infinity norm (alpha channel is not computed).*

- `NppStatus nppiNorm_Inf_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit signed short image infinity norm (alpha channel is not computed).*

- `NppStatus nppiNorm_Inf_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)  
*4-channel 32-bit floating-point image infinity norm (alpha channel is not computed).*
- `NppStatus nppiNorm_Inf_8u_C4R` (const `Npp8u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 8-bit unsigned char image infinity norm.*
- `NppStatus nppiNorm_Inf_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit unsigned short image infinity norm.*
- `NppStatus nppiNorm_Inf_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit signed short image infinity norm.*
- `NppStatus nppiNorm_Inf_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 32-bit floating-point image infinity norm.*
- `NppStatus nppiNorm_Inf_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit unsigned char image infinity norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_Inf_8s_C3CMR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit signed char image infinity norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_Inf_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 16-bit unsigned short image infinity norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_Inf_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 32-bit floating-point image infinity norm, Masked Operation, Channel-of-Interest API.*

## 7.82.1 Function Documentation

### 7.82.1.1 `NppStatus nppiNorm_Inf_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u *pDeviceBuffer`)

4-channel 16-bit signed short image infinity norm (alpha channel is not computed).

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.82.1.2 NppStatus nppiNorm\_Inf\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

1-channel 16-bit signed short image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.82.1.3 NppStatus nppiNorm\_Inf\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)

3-channel 16-bit signed short image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.82.1.4 NppStatus nppiNorm\_Inf\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)

4-channel 16-bit signed short image infinity norm.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNorm* Array that contains the norm values of four channels.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.82.1.5 NppStatus nppiNorm\_Inf\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)

4-channel 16-bit unsigned short image infinity norm (alpha channel is not computed).

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- aNorm* Array that contains the norm values of three channels.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.82.1.6 NppStatus nppiNorm\_Inf\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

1-channel 16-bit unsigned short image infinity norm, [Masked Operation](#).

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNorm* Pointer to the norm value.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.7** `NppStatus nppiNorm_Inf_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 16-bit unsigned short image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.8** `NppStatus nppiNorm_Inf_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image infinity norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.82.1.9** `NppStatus nppiNorm_Inf_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.10 NppStatus nppiNorm\_Inf\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

4-channel 16-bit unsigned short image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.11 NppStatus nppiNorm\_Inf\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

4-channel 32-bit floating-point image infinity norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.12** `NppStatus nppiNorm_Inf_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating-point image infinity norm, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.13** `NppStatus nppiNorm_Inf_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating-point image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.14** `NppStatus nppiNorm_Inf_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 32-bit floating-point image infinity norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.82.1.15** `NppStatus nppiNorm_Inf_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 32-bit floating-point image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.16** `NppStatus nppiNorm_Inf_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating-point image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.17** `NppStatus nppiNorm_Inf_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 32-bit signed int image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.18** `NppStatus nppiNorm_Inf_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit signed char image infinity norm, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.19** `NppStatus nppiNorm_Inf_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit signed char image infinity norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.82.1.20** `NppStatus nppiNorm_Inf_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned char image infinity norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.21** `NppStatus nppiNorm_Inf_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image infinity norm, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.22** `NppStatus nppiNorm_Inf_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.23** `NppStatus nppiNorm_Inf_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned char image infinity norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.82.1.24** `NppStatus nppiNorm_Inf_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned char image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.25** `NppStatus nppiNorm_Inf_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned char image infinity norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.82.1.26** `NppStatus nppiNormInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormInf_16s_AC4R`.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.27** `NppStatus nppiNormInfGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormInf_16s_C1R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.28 NppStatus nppiNormInfGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for `nppiNormInf_16s_C3R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.29 NppStatus nppiNormInfGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for `nppiNormInf_16s_C4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.30 NppStatus nppiNormInfGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for `nppiNormInf_16u_AC4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.31 NppStatus nppiNormInfGetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.32 NppStatus nppiNormInfGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.33 NppStatus nppiNormInfGetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.34 NppStatus nppiNormInfGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.35 NppStatus nppiNormInfGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for `nppiNormInf_16u_C4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.36 NppStatus nppiNormInfGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for `nppiNormInf_32f_AC4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.37 NppStatus nppiNormInfGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for `nppiNormInf_32f_C1MR`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.38 NppStatus nppiNormInfGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.39 NppStatus nppiNormInfGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.40 NppStatus nppiNormInfGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.41 NppStatus nppiNormInfGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormInf\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.42** `NppStatus nppiNormInfGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormInf_32s_C1R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.43** `NppStatus nppiNormInfGetBufferHostSize_8s_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormInf_8s_C1MR`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.44** `NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormInf_8s_C3CMR`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.45 NppStatus nppiNormInfGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.46 NppStatus nppiNormInfGetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.47 NppStatus nppiNormInfGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.48 NppStatus nppiNormInfGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.49** NppStatus nppiNormInfGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.82.1.50** NppStatus nppiNormInfGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormInf\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.83 L1 Norm

### L1 Norm

These functions compute the L1 norm of an image.

The L1 norm is defined as the sum of all the absolute pixel values in the image. If the image contains multiple channels, the functions will compute the norm for each channel separately. The functions require the addition device scratch buffer for the computations.

- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C1R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C1R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16s\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C1R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C1R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C1R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C1MR.*
- [NppStatus nppiNormL1GetBufferHostSize\\_8s\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8s\_C1MR.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C1MR.*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C1MR](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C1MR.*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C3R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C3R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C3R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C3R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_AC4R.*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_AC4R](#) (NppiSize oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_AC4R.*

- **NppStatus nppiNormL1GetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16s\_AC4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_AC4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C4R.*
- **NppStatus nppiNormL1GetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C3CMR.*
- **NppStatus nppiNormL1GetBufferHostSize\_8s\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_8s\_C3CMR.*
- **NppStatus nppiNormL1GetBufferHostSize\_16u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C3CMR.*
- **NppStatus nppiNormL1GetBufferHostSize\_32f\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C3CMR.*
- **NppStatus nppiNorm\_L1\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 8-bit unsigned char image L1 norm.*
- **NppStatus nppiNorm\_L1\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit unsigned short image L1 norm.*
- **NppStatus nppiNorm\_L1\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit signed short image L1 norm.*
- **NppStatus nppiNorm\_L1\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 32-bit floating-point image L1 norm.*
- **NppStatus nppiNorm\_L1\_8u\_C1MR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)

*1-channel 8-bit unsigned char image L1 norm, Masked Operation*

- `NppStatus nppiNorm_L1_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 8-bit signed char image L1 norm, Masked Operation*

- `NppStatus nppiNorm_L1_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 16-bit unsigned short image L1 norm, Masked Operation*

- `NppStatus nppiNorm_L1_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*1-channel 32-bit floating-point image L1 norm, Masked Operation.*

- `NppStatus nppiNorm_L1_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit unsigned char image L1 norm.*

- `NppStatus nppiNorm_L1_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit unsigned short image L1 norm.*

- `NppStatus nppiNorm_L1_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit signed short image L1 norm.*

- `NppStatus nppiNorm_L1_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 32-bit floating-point image L1 norm.*

- `NppStatus nppiNorm_L1_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 8-bit unsigned char image L1 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L1_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit unsigned short image L1 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L1_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit signed short image L1 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L1_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 32-bit floating-point image L1 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L1_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*4-channel 8-bit unsigned char image L1 norm.*

- `NppStatus nppiNorm_L1_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit unsigned short image L1 norm.*
- `NppStatus nppiNorm_L1_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit signed short image L1 norm.*
- `NppStatus nppiNorm_L1_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 32-bit floating-point image L1 norm.*
- `NppStatus nppiNorm_L1_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit unsigned char image L1 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L1_8s_C3CMR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit signed char image L1 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L1_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 16-bit unsigned short image L1 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L1_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 32-bit floating-point image L1 norm, Masked Operation, Channel-of-Interest API.*

### 7.83.1 Function Documentation

#### 7.83.1.1 `NppStatus nppiNorm_L1_16s_AC4R` (const `Npp16s * pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u * pDeviceBuffer`)

4-channel 16-bit signed short image L1 norm (alpha channel is not computed).

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppiNormL1GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.83.1.2 `NppStatus nppiNorm_L1_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 16-bit signed short image L1 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.83.1.3 `NppStatus nppiNorm_L1_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 16-bit signed short image L1 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.83.1.4 `NppStatus nppiNorm_L1_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 16-bit signed short image L1 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.5 NppStatus nppiNorm\_L1\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

4-channel 16-bit unsigned short image L1 norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.6 NppStatus nppiNorm\_L1\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

1-channel 16-bit unsigned short image L1 norm, [Masked Operation](#)

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.7 NppStatus nppiNorm\_L1\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

1-channel 16-bit unsigned short image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.8** `NppStatus nppiNorm_L1_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image L1 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.83.1.9** `NppStatus nppiNorm_L1_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image L1 norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.10** `NppStatus nppiNorm_L1_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 16-bit unsigned short image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.11** `NppStatus nppiNorm_L1_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating-point image L1 norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.12** `NppStatus nppiNorm_L1_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 32-bit floating-point image L1 norm, [Masked Operation](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.83.1.13 NppStatus nppiNorm\_L1\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

1-channel 32-bit floating-point image L1 norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.83.1.14 NppStatus nppiNorm\_L1\_32f\_C3CMR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

3-channel 32-bit floating-point image L1 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [NPP\\_NOT\\_EVEN\\_STEP\\_ERROR](#) if the step of the source image cannot be divided by 4, or [NPP\\_COI\\_ERROR](#) if an invalid channel of interest is specified.

**7.83.1.15** `NppStatus nppiNorm_L1_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 32-bit floating-point image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.16** `NppStatus nppiNorm_L1_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating-point image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.17** `NppStatus nppiNorm_L1_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit signed char image L1 norm, [Masked Operation](#)

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.18** `NppStatus nppiNorm_L1_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit signed char image L1 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.83.1.19** `NppStatus nppiNorm_L1_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned char image L1 norm (alpha channel is not computed).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.20** `NppStatus nppiNorm_L1_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image L1 norm, [Masked Operation](#)

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.21** `NppStatus nppiNorm_L1_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.22** `NppStatus nppiNorm_L1_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned char image L1 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.83.1.23 NppStatus nppiNorm\_L1\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

3-channel 8-bit unsigned char image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.24 NppStatus nppiNorm\_L1\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

4-channel 8-bit unsigned char image L1 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.83.1.25 NppStatus nppiNormL1GetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.26 NppStatus nppiNormL1GetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.27 NppStatus nppiNormL1GetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.28 NppStatus nppiNormL1GetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.29** NppStatus nppiNormL1GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.30** NppStatus nppiNormL1GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.31** NppStatus nppiNormL1GetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.32 NppStatus nppiNormL1GetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.33 NppStatus nppiNormL1GetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.34 NppStatus nppiNormL1GetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.35 NppStatus nppiNormL1GetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.36** NppStatus nppiNormL1GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.37** NppStatus nppiNormL1GetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.38** NppStatus nppiNormL1GetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.39 NppStatus nppiNormL1GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.40 NppStatus nppiNormL1GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.41 NppStatus nppiNormL1GetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.42 NppStatus nppiNormL1GetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.43 NppStatus nppiNormL1GetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiNormL1\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.44 NppStatus nppiNormL1GetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiNormL1\_8u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.45 NppStatus nppiNormL1GetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiNormL1\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.46 NppStatus nppiNormL1GetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.47 NppStatus nppiNormL1GetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.83.1.48 NppStatus nppiNormL1GetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL1\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.84 L2 Norm

### L2 Norm

These functions compute the L2 norm of an image.

The L2 norm is defined as the sum of all the square pixel values in the image. If the image contains multiple channels, the functions will compute the norm for each channel separately. The functions require the addition device scratch buffer for the computations.

- [NppStatus nppiNormL2GetBufferHostSize\\_8u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C1R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C1R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16s\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C1R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C1R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_8u\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C1MR.*
- [NppStatus nppiNormL2GetBufferHostSize\\_8s\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8s\_C1MR.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16u\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C1MR.*
- [NppStatus nppiNormL2GetBufferHostSize\\_32f\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C1MR.*
- [NppStatus nppiNormL2GetBufferHostSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C3R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C3R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16s\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C3R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C3R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_AC4R.*
- [NppStatus nppiNormL2GetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_AC4R.*

- **NppStatus nppiNormL2GetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16s\_AC4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_AC4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C4R.*
- **NppStatus nppiNormL2GetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C3CMR.*
- **NppStatus nppiNormL2GetBufferHostSize\_8s\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_8s\_C3CMR.*
- **NppStatus nppiNormL2GetBufferHostSize\_16u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C3CMR.*
- **NppStatus nppiNormL2GetBufferHostSize\_32f\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C3CMR.*
- **NppStatus nppiNorm\_L2\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 8-bit unsigned char image L2 norm.*
- **NppStatus nppiNorm\_L2\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit unsigned short image L2 norm.*
- **NppStatus nppiNorm\_L2\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 16-bit signed short image L2 norm.*
- **NppStatus nppiNorm\_L2\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*1-channel 32-bit floating-point image L2 norm.*
- **NppStatus nppiNorm\_L2\_8u\_C1MR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)

*Masked Operation 1-channel 8-bit unsigned char image L2 norm.*

- `NppStatus nppiNorm_L2_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked Operation 1-channel 8-bit signed char image L2 norm.*

- `NppStatus nppiNorm_L2_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked Operation 1-channel 16-bit unsigned short image L2 norm.*

- `NppStatus nppiNorm_L2_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked Operation 1-channel 32-bit floating-point image L2 norm.*

- `NppStatus nppiNorm_L2_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 8-bit unsigned char image L2 norm.*

- `NppStatus nppiNorm_L2_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit unsigned short image L2 norm.*

- `NppStatus nppiNorm_L2_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 16-bit signed short image L2 norm.*

- `NppStatus nppiNorm_L2_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*3-channel 32-bit floating-point image L2 norm.*

- `NppStatus nppiNorm_L2_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 8-bit unsigned char image L2 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L2_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit unsigned short image L2 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L2_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 16-bit signed short image L2 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L2_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*4-channel 32-bit floating-point image L2 norm (alpha channel is not computed).*

- `NppStatus nppiNorm_L2_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*4-channel 8-bit unsigned char image L2 norm.*

- `NppStatus nppiNorm_L2_16u_C4R` (const `Npp16u *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit unsigned short image L2 norm.*
- `NppStatus nppiNorm_L2_16s_C4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 16-bit signed short image L2 norm.*
- `NppStatus nppiNorm_L2_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[4]`, `Npp8u *pDeviceBuffer`)  
*4-channel 32-bit floating-point image L2 norm.*
- `NppStatus nppiNorm_L2_8u_C3CMR` (const `Npp8u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit unsigned char image L2 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L2_8s_C3CMR` (const `Npp8s *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 8-bit signed char image L2 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L2_16u_C3CMR` (const `Npp16u *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 16-bit unsigned short image L2 norm, Masked Operation, Channel-of-Interest API.*
- `NppStatus nppiNorm_L2_32f_C3CMR` (const `Npp32f *pSrc`, int `nSrcStep`, const `Npp8u *pMask`, int `nMaskStep`, `NppiSize oSizeROI`, int `nCOI`, `Npp64f *pNorm`, `Npp8u *pDeviceBuffer`)  
*3-channel 32-bit floating-point image L2 norm, Masked Operation, Channel-of-Interest API.*

## 7.84.1 Function Documentation

### 7.84.1.1 `NppStatus nppiNorm_L2_16s_AC4R` (const `Npp16s * pSrc`, int `nSrcStep`, `NppiSize oSizeROI`, `Npp64f aNorm[3]`, `Npp8u * pDeviceBuffer`)

4-channel 16-bit signed short image L2 norm (alpha channel is not computed).

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppiNormL2GetBufferHostSize_16s_AC4R` to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.84.1.2 `NppStatus nppiNorm_L2_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 16-bit signed short image L2 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.84.1.3 `NppStatus nppiNorm_L2_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 16-bit signed short image L2 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.84.1.4 `NppStatus nppiNorm_L2_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 16-bit signed short image L2 norm.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.5 NppStatus nppiNorm\_L2\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

4-channel 16-bit unsigned short image L2 norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.6 NppStatus nppiNorm\_L2\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

[Masked Operation](#) 1-channel 16-bit unsigned short image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.7 NppStatus nppiNorm\_L2\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

1-channel 16-bit unsigned short image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.8** `NppStatus nppiNorm_L2_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image L2 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.84.1.9** `NppStatus nppiNorm_L2_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 16-bit unsigned short image L2 norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.10** `NppStatus nppiNorm_L2_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 16-bit unsigned short image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.11** `NppStatus nppiNorm_L2_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating-point image L2 norm (alpha channel is not computed).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.12** `NppStatus nppiNorm_L2_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

[Masked Operation](#) 1-channel 32-bit floating-point image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4.

**7.84.1.13 NppStatus nppiNorm\_L2\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

1-channel 32-bit floating-point image L2 norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.14 NppStatus nppiNorm\_L2\_32f\_C3CMR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

3-channel 32-bit floating-point image L2 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.84.1.15** `NppStatus nppiNorm_L2_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

3-channel 32-bit floating-point image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.16** `NppStatus nppiNorm_L2_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

4-channel 32-bit floating-point image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.17** `NppStatus nppiNorm_L2_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

[Masked Operation](#) 1-channel 8-bit signed char image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.18** `NppStatus nppiNorm_L2_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit signed char image L2 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.84.1.19** `NppStatus nppiNorm_L2_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

4-channel 8-bit unsigned char image L2 norm (alpha channel is not computed).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.20** `NppStatus nppiNorm_L2_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

[Masked Operation](#) 1-channel 8-bit unsigned char image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.21** `NppStatus nppiNorm_L2_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

1-channel 8-bit unsigned char image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.22** `NppStatus nppiNorm_L2_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

3-channel 8-bit unsigned char image L2 norm, [Masked Operation](#), [Channel-of-Interest API](#).

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.84.1.23 NppStatus nppiNorm\_L2\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

3-channel 8-bit unsigned char image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of three channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.24 NppStatus nppiNorm\_L2\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

4-channel 8-bit unsigned char image L2 norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of four channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.84.1.25 NppStatus nppiNormL2GetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.26 NppStatus nppiNormL2GetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.27 NppStatus nppiNormL2GetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.28 NppStatus nppiNormL2GetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.29** NppStatus nppiNormL2GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.30** NppStatus nppiNormL2GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.31** NppStatus nppiNormL2GetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.32 NppStatus nppiNormL2GetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.33 NppStatus nppiNormL2GetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.34 NppStatus nppiNormL2GetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.35 NppStatus nppiNormL2GetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.36** NppStatus nppiNormL2GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.37** NppStatus nppiNormL2GetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.38** NppStatus nppiNormL2GetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.39 NppStatus nppiNormL2GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.40 NppStatus nppiNormL2GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.41 NppStatus nppiNormL2GetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.42 NppStatus nppiNormL2GetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.43** `NppStatus nppiNormL2GetBufferHostSize_8u_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormL2_8u_AC4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.44** `NppStatus nppiNormL2GetBufferHostSize_8u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormL2_8u_C1MR`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.45** `NppStatus nppiNormL2GetBufferHostSize_8u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppiNormL2_8u_C1R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.46 NppStatus nppiNormL2GetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C3CMR.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.47 NppStatus nppiNormL2GetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C3R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.84.1.48 NppStatus nppiNormL2GetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiNormL2\_8u\_C4R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.85 Norm Diff

### NormDiff

Norm of pixel differences between two images.

- `NppStatus nppiNormDiff_L1_8u_C1R` (const `Npp8u *pSrc1`, int `nSrcStep1`, const `Npp8u *pSrc2`, int `nSrcStep2`, `NppiSize oSizeROI`, `Npp64f *pRetVal`)  
*8-bit unsigned L1 norm of pixel differences.*
- `NppStatus nppiNormDiff_L2_8u_C1R` (const `Npp8u *pSrc1`, int `nSrcStep1`, const `Npp8u *pSrc2`, int `nSrcStep2`, `NppiSize oSizeROI`, `Npp64f *pRetVal`)  
*8-bit unsigned L2 norm of pixel differences.*
- `NppStatus nppiNormDiff_Inf_8u_C1R` (const `Npp8u *pSrc1`, int `nSrcStep1`, const `Npp8u *pSrc2`, int `nSrcStep2`, `NppiSize oSizeROI`, `Npp64f *pRetVal`)  
*8-bit unsigned Infinity Norm of pixel differences.*

### 7.85.1 Function Documentation

#### 7.85.1.1 `NppStatus nppiNormDiff_Inf_8u_C1R` (const `Npp8u *pSrc1`, int `nSrcStep1`, const `Npp8u *pSrc2`, int `nSrcStep2`, `NppiSize oSizeROI`, `Npp64f *pRetVal`)

8-bit unsigned Infinity Norm of pixel differences.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrcStep1* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrcStep2* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*\*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.85.1.2 `NppStatus nppiNormDiff_L1_8u_C1R` (const `Npp8u *pSrc1`, int `nSrcStep1`, const `Npp8u *pSrc2`, int `nSrcStep2`, `NppiSize oSizeROI`, `Npp64f *pRetVal`)

8-bit unsigned L1 norm of pixel differences.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrcStep1* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrcStep2* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.85.1.3 NppStatus nppiNormDiff\_L2\_8u\_C1R (const Npp8u \* pSrc1, int nSrcStep1, const Npp8u \* pSrc2, int nSrcStep2, NppiSize oSizeROI, Npp64f \* pRetVal)**

8-bit unsigned L2 norm of pixel differences.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrcStep1* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrcStep2* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pRetVal* Contains computed L1-norm of differences. This is a host pointer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.86 Integral and Rectangular Standard Deviation

### Integral

- `NppStatus nppiSqrIntegral_8u32s32f_C1R` (`Npp8u *pSrc`, `int nSrcStep`, `Npp32s *pDst`, `int nDstStep`, `Npp32f *pSqr`, `int nSqrStep`, `NppiSize oSrcROI`, `Npp32s val`, `Npp32f valSqr`, `Npp32s integralImageNewHeight`)

*SqrIntegral* Transforms an image to integral and integral of pixel squares representation.

- `NppStatus nppiRectStdDev_32s32f_C1R` (`const Npp32s *pSrc`, `int nSrcStep`, `const Npp64f *pSqr`, `int nSqrStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `NppiRect rect`)

*RectStdDev* Computes the standard deviation of integral images.

### 7.86.1 Function Documentation

- 7.86.1.1** `NppStatus nppiRectStdDev_32s32f_C1R` (`const Npp32s *pSrc`, `int nSrcStep`, `const Npp64f *pSqr`, `int nSqrStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `NppiRect rect`)

*RectStdDev* Computes the standard deviation of integral images.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSqr* Destination-Image Pointer.  
*nSqrStep* Destination-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rect* rectangular window

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- 7.86.1.2** `NppStatus nppiSqrIntegral_8u32s32f_C1R` (`Npp8u *pSrc`, `int nSrcStep`, `Npp32s *pDst`, `int nDstStep`, `Npp32f *pSqr`, `int nSqrStep`, `NppiSize oSrcROI`, `Npp32s val`, `Npp32f valSqr`, `Npp32s integralImageNewHeight`)

*SqrIntegral* Transforms an image to integral and integral of pixel squares representation.

This function assumes that the integral and integral of squares images.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pSqr* Destination-Image Pointer.

*nSqrStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

*val* The value to add to pDst image pixels

*valSqr* The value to add to pSqr image pixels

*integralImageNewHeight* Extended height of output surfaces (needed by transpose in primitive)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.87 Histogram

### Histogram

- [NppStatus nppiEvenLevelsHost\\_32s](#) ([Npp32s](#) \*hpLevels, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel)  
*Compute levels with even distribution.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_8u\\_C1R](#) ([NppiSize](#) oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_C1R.*
- [NppStatus nppiHistogramEven\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) \*pBuffer)  
*8-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_C3R.*
- [NppStatus nppiHistogramEven\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*3 channel 8-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_C4R.*
- [NppStatus nppiHistogramEven\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) \*pBuffer)  
*4 channel 8-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_8u\_AC4R.*
- [NppStatus nppiHistogramEven\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*4 channel (alpha as the last channel) 8-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16u\\_C1R](#) ([NppiSize](#) oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_C1R.*
- [NppStatus nppiHistogramEven\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) \*pBuffer)  
*16-bit unsigned histogram with evenly distributed bins.*

- [NppStatus nppiHistogramEvenGetBufferSize\\_16u\\_C3R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_C3R.*
- [NppStatus nppiHistogramEven\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*3 channel 16-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_C4R.*
- [NppStatus nppiHistogramEven\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) \*pBuffer)  
*4 channel 16-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16u\_AC4R.*
- [NppStatus nppiHistogramEven\\_16u\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*4 channel (alpha as the last channel) 16-bit unsigned histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16s\\_C1R](#) ([NppiSize](#) oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_C1R.*
- [NppStatus nppiHistogramEven\\_16s\\_C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) \*pBuffer)  
*16-bit signed histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16s\\_C3R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_C3R.*
- [NppStatus nppiHistogramEven\\_16s\\_C3R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*3 channel 16-bit signed histogram with evenly distributed bins.*
- [NppStatus nppiHistogramEvenGetBufferSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_C4R.*
- [NppStatus nppiHistogramEven\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) \*pBuffer)

*4 channel 16-bit signed histogram with evenly distributed bins.*

- [NppStatus nppiHistogramEvenGetBufferSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramEven\_16s\_AC4R.*
- [NppStatus nppiHistogramEven\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*4 channel (alpha as the last channel) 16-bit signed histogram with evenly distributed bins.*
- [NppStatus nppiHistogramRangeGetBufferSize\\_8u\\_C1R](#) ([NppiSize](#) oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C1R.*
- [NppStatus nppiHistogramRange\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, const [Npp32s](#) \*pLevels, int nLevels, [Npp8u](#) \*pBuffer)  
*8-bit unsigned histogram with bins determined by pLevels array.*
- [NppStatus nppiHistogramRangeGetBufferSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C3R.*
- [NppStatus nppiHistogramRange\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], const [Npp32s](#) \*pLevels[3], int nLevels[3], [Npp8u](#) \*pBuffer)  
*3 channel 8-bit unsigned histogram with bins determined by pLevels.*
- [NppStatus nppiHistogramRangeGetBufferSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C4R.*
- [NppStatus nppiHistogramRange\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], const [Npp32s](#) \*pLevels[4], int nLevels[4], [Npp8u](#) \*pBuffer)  
*4 channel 8-bit unsigned histogram with bins determined by pLevels.*
- [NppStatus nppiHistogramRangeGetBufferSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.*
- [NppStatus nppiHistogramRange\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], const [Npp32s](#) \*pLevels[3], int nLevels[3], [Npp8u](#) \*pBuffer)  
*4 channel (alpha as a last channel) 8-bit unsigned histogram with bins determined by pLevels.*
- [NppStatus nppiHistogramRangeGetBufferSize\\_16u\\_C1R](#) ([NppiSize](#) oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C1R.*
- [NppStatus nppiHistogramRange\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, const [Npp32s](#) \*pLevels, int nLevels, [Npp8u](#) \*pBuffer)  
*16-bit unsigned histogram with bins determined by pLevels array.*

- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_C3R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C3R.*
- **NppStatus** `nppiHistogramRange_16u_C3R` (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*3 channel 16-bit unsigned histogram with bins determined by pLevels.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_C4R` (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C4R.*
- **NppStatus** `nppiHistogramRange_16u_C4R` (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*4 channel 16-bit unsigned histogram with bins determined by pLevels.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16u_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.*
- **NppStatus** `nppiHistogramRange_16u_AC4R` (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*4 channel (alpha as a last channel) 16-bit unsigned histogram with bins determined by pLevels.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C1R` (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C1R.*
- **NppStatus** `nppiHistogramRange_16s_C1R` (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*16-bit signed histogram with bins determined by pLevels array.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C3R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C3R.*
- **NppStatus** `nppiHistogramRange_16s_C3R` (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*3 channel 16-bit signed histogram with bins determined by pLevels.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_C4R` (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C4R.*
- **NppStatus** `nppiHistogramRange_16s_C4R` (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*4 channel 16-bit signed histogram with bins determined by pLevels.*
- **NppStatus** `nppiHistogramRangeGetBufferSize_16s_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_AC4R.*

- `NppStatus nppiHistogramRange_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32s *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)  
*4 channel (alpha as a last channel) 16-bit signed histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_32f_C1R` (`NppiSize` `oSizeROI`, int `nLevels`, int `*hpBufferSize`)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C1R.*
- `NppStatus nppiHistogramRange_32f_C1R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist`, const `Npp32f *pLevels`, int `nLevels`, `Npp8u *pBuffer`)  
*32-bit float histogram with bins determined by pLevels array.*
- `NppStatus nppiHistogramRangeGetBufferSize_32f_C3R` (`NppiSize` `oSizeROI`, int `nLevels[3]`, int `*hpBufferSize`)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C3R.*
- `NppStatus nppiHistogramRange_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32f *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)  
*3 channel 32-bit float histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_32f_C4R` (`NppiSize` `oSizeROI`, int `nLevels[4]`, int `*hpBufferSize`)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C4R.*
- `NppStatus nppiHistogramRange_32f_C4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[4]`, const `Npp32f *pLevels[4]`, int `nLevels[4]`, `Npp8u *pBuffer`)  
*4 channel 32-bit float histogram with bins determined by pLevels.*
- `NppStatus nppiHistogramRangeGetBufferSize_32f_AC4R` (`NppiSize` `oSizeROI`, int `nLevels[3]`, int `*hpBufferSize`)  
*Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.*
- `NppStatus nppiHistogramRange_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `NppiSize` `oSizeROI`, `Npp32s *pHist[3]`, const `Npp32f *pLevels[3]`, int `nLevels[3]`, `Npp8u *pBuffer`)  
*4 channel (alpha as a last channel) 32-bit float histogram with bins determined by pLevels.*

## 7.87.1 Function Documentation

### 7.87.1.1 `NppStatus nppiEvenLevelsHost_32s` (`Npp32s *hpLevels`, int `nLevels`, `Npp32s nLowerLevel`, `Npp32s nUpperLevel`)

Compute levels with even distribution.

#### Parameters:

***hpLevels*** A host pointer to array which receives the levels being computed. The array needs to be of size `nLevels`.

***nLevels*** The number of levels being computed. `nLevels` must be at least 2, otherwise an `NPP_HISTO_NUMBER_OF_LEVELS_ERROR` error is returned.

*nLowerLevel* Lower boundary value of the lowest level.

*nUpperLevel* Upper boundary value of the greatest level.

**Returns:**

Error code.

**7.87.1.2 NppStatus nppiHistogramEven\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u \* pBuffer)**

4 channel (alpha as the last channel) 16-bit signed histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16s\_AC4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.3 NppStatus nppiHistogramEven\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u \* pBuffer)**

16-bit signed histogram with evenly distributed bins.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (nppiHistogramEvenGetBufferSize\_16s\_C1R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.4 NppStatus nppiHistogramEven\_16s\_C3R** (*const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u \* pBuffer*)

3 channel 16-bit signed histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (*nppiHistogramEvenGetBufferSize\_16s\_C3R*) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.5 NppStatus nppiHistogramEven\_16s\_C4R** (*const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u \* pBuffer*)

4 channel 16-bit signed histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (*nppiHistogramEvenGetBufferSize\_16s\_C4R*) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.6** `NppStatus nppiHistogramEven_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

4 channel (alpha as the last channel) 16-bit unsigned histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_16u_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.7** `NppStatus nppiHistogramEven_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

16-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_16u_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.8 NppStatus nppiHistogramEven\_16u\_C3R** (`const Npp16u * pSrc`, `int nSrcStep`, `NppiSize oSizeROI`, `Npp32s * pHist[3]`, `int nLevels[3]`, `Npp32s nLowerLevel[3]`, `Npp32s nUpperLevel[3]`, `Npp8u * pBuffer`)

3 channel 16-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]*-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_16u_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.9 NppStatus nppiHistogramEven\_16u\_C4R** (`const Npp16u * pSrc`, `int nSrcStep`, `NppiSize oSizeROI`, `Npp32s * pHist[4]`, `int nLevels[4]`, `Npp32s nLowerLevel[4]`, `Npp32s nUpperLevel[4]`, `Npp8u * pBuffer`)

4 channel 16-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]*-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_16u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.10** `NppStatus nppiHistogramEven_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

4 channel (alpha as the last channel) 8-bit unsigned histogram with evenly distributed bins.

Alpha channel is ignored during histogram computation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.11** `NppStatus nppiHistogramEven_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

8-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.12** `NppStatus nppiHistogramEven_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

3 channel 8-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.13** `NppStatus nppiHistogramEven_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

4 channel 8-bit unsigned histogram with evenly distributed bins.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized (`nppiHistogramEvenGetBufferSize_8u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.14 NppStatus nppiHistogramEvenGetBufferSize\_16s\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16s\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.15 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.16 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16s\_C3R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.17 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16s\_C4R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.18 NppStatus nppiHistogramEvenGetBufferSize\_16u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.19 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nLevels* Number of levels in the histogram.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.20 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16u\_C3R.

**Parameters:**

*oSizeROI* ROI size.  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.21 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_16u\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.22 NppStatus nppiHistogramEvenGetBufferSize\_8u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_8u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.23 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.24 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramEven\_8u\_C3R.

**Parameters:**

- oSizeROI* [Region-of-Interest \(ROI\)](#).
- nLevels* Number of levels in the histogram.
- hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.87.1.25 `NppStatus nppiHistogramEvenGetBufferSize_8u_C4R (NppiSize oSizeROI, int nLevels[4], int * hpBufferSize)`

Scratch-buffer size for `nppiHistogramEven_8u_C4R`.

**Parameters:**

- oSizeROI* ROI size.
- nLevels* Array containing number of levels per color channel.
- hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

#### 7.87.1.26 `NppStatus nppiHistogramRange_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

4 channel (alpha as a last channel) 16-bit signed histogram with bins determined by `pLevels`.

Alpha channel is ignored during the histograms computations.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by `pHist[i]` must be of size `nLevels[i]-1`.
- nLevels* Array containing number of levels per color channel.
- pLevels* Array containing pointers to level-arrays per color channel. Array pointed by `pLevel[i]` must be of size `nLevels[i]`.
- pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.27** `NppStatus nppiHistogramRange_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

16-bit signed histogram with bins determined by pLevels array.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size nLevels.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.28** `NppStatus nppiHistogramRange_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

3 channel 16-bit signed histogram with bins determined by pLevels.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by `pHist[i]` must be of size `nLevels[i]-1`.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by `pLevel[i]` must be of size `nLevels[i]`.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.29** `NppStatus nppiHistogramRange_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

4 channel 16-bit signed histogram with bins determined by pLevels.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16s_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.30** `NppStatus nppiHistogramRange_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

4 channel (alpha as a last channel) 16-bit unsigned histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16u_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.31** `NppStatus nppiHistogramRange_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

16-bit unsigned histogram with bins determined by *pLevels* array.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16u_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.32** `NppStatus nppiHistogramRange_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

3 channel 16-bit unsigned histogram with bins determined by *pLevels*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16u_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.33** `NppStatus nppiHistogramRange_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

4 channel 16-bit unsigned histogram with bins determined by *pLevels*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_16u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.34** `NppStatus nppiHistogramRange_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

4 channel (alpha as a last channel) 32-bit float histogram with bins determined by *pLevels*.

Alpha channel is ignored during the histograms computations.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.35** `NppStatus nppiHistogramRange_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32f * pLevels, int nLevels, Npp8u * pBuffer)`

32-bit float histogram with bins determined by *pLevels* array.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels-1*.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.36** `NppStatus nppiHistogramRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32f * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

3 channel 32-bit float histogram with bins determined by pLevels.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.37** `NppStatus nppiHistogramRange_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32f * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

4 channel 32-bit float histogram with bins determined by pLevels.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.38** `NppStatus nppiHistogramRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

4 channel (alpha as a last channel) 8-bit unsigned histogram with bins determined by pLevels.

Alpha channel is ignored during the histograms computations.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.87.1.39 `NppStatus nppiHistogramRange_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

8-bit unsigned histogram with bins determined by *pLevels* array.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels-1*.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_C1R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.87.1.40 `NppStatus nppiHistogramRange_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

3 channel 8-bit unsigned histogram with bins determined by *pLevels*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_C3R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.41** `NppStatus nppiHistogramRange_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

4 channel 8-bit unsigned histogram with bins determined by *pLevels*.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist[i]* must be of size *nLevels[i]-1*.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel[i]* must be of size *nLevels[i]*.

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.87.1.42** `NppStatus nppiHistogramRangeGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)`

Scratch-buffer size for `nppiHistogramRange_16s_AC4R`.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.43 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16s\_C1R.

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.44 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16s\_C3R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.45 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16s\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.46 NppStatus nppiHistogramRangeGetBufferSize\_16u\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.47 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.48 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16u\_C3R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.49 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_16u\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.50 NppStatus nppiHistogramRangeGetBufferSize\_32f\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.51 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.52 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_C3R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.53 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.54 NppStatus nppiHistogramRangeGetBufferSize\_8u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.55 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.56 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_C3R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

**7.87.1.57 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_8u\_C4R.

**Parameters:**

*oSizeROI* ROI size.

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Host pointer where required buffer size is returned.

**Returns:**

Error Code.

## 7.88 Memory Management

Routines for allocating and deallocating pitched image storage.

### Functions

- void [nppiFree](#) (void \*pData)  
*Free method for any 2D allocated memory.*

### Image Memory Allocation

ImageAllocator methods for 2D arrays of data.

The allocators have width and height parameters to specify the size of the image data being allocated. They return a pointer to the newly created memory and return the numbers of bytes between successive lines.

If the memory allocation failed due to lack of free device memory or device memory fragmentation the routine returns 0.

All allocators return memory with line strides that are beneficial for performance. It is not mandatory to use these allocators. Any valid CUDA device-memory pointers can be used by the NPP primitives and there are no restrictions on line strides.

- $\text{Npp8u} * \text{nppiMalloc\_8u\_C1}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*8-bit unsigned image memory allocator.*
- $\text{Npp8u} * \text{nppiMalloc\_8u\_C2}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 8-bit unsigned image memory allocator.*
- $\text{Npp8u} * \text{nppiMalloc\_8u\_C3}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 8-bit unsigned image memory allocator.*
- $\text{Npp8u} * \text{nppiMalloc\_8u\_C4}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 8-bit unsigned image memory allocator.*
- $\text{Npp16u} * \text{nppiMalloc\_16u\_C1}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit unsigned image memory allocator.*
- $\text{Npp16u} * \text{nppiMalloc\_16u\_C2}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit unsigned image memory allocator.*
- $\text{Npp16u} * \text{nppiMalloc\_16u\_C3}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit unsigned image memory allocator.*
- $\text{Npp16u} * \text{nppiMalloc\_16u\_C4}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit unsigned image memory allocator.*
- $\text{Npp16s} * \text{nppiMalloc\_16s\_C1}$  (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit signed image memory allocator.*

- `Npp16s * nppiMalloc_16s_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed image memory allocator.*
- `Npp16s * nppiMalloc_16s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*1 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed complex image memory allocator.*
- `Npp32s * nppiMalloc_32s_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit signed image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit integer complex image memory allocator.*
- `Npp32f * nppiMalloc_32f_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit floating point image memory allocator.*

- `Npp32fc * nppiMalloc_32fc_C1` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C2` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*2 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C3` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*3 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C4` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*4 channel 32-bit float complex image memory allocator.*

### 7.88.1 Detailed Description

Routines for allocating and deallocating pitched image storage.

These methods are provided for convenience. They allocate memory that may contain additional padding bytes at the end of each line of pixels. Though padding is not necessary for any of the NPP image-processing primitives to work correctly, its absence may cause severe performance degradation compared to properly padded images.

### 7.88.2 Function Documentation

#### 7.88.2.1 void nppiFree (void \* *pData*)

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the `nppiMalloc_<modifier>` methods.

#### Parameters:

*pData* A pointer to memory allocated using `nppiMalloc_<modifier>`.

#### 7.88.2.2 Npp16s\* nppiMalloc\_16s\_C1 (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)

16-bit signed image memory allocator.

#### Parameters:

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* Line Step.

#### Returns:

Pointer to new image data.

**7.88.2.3 Npp16s\* nppiMalloc\_16s\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.4 Npp16s\* nppiMalloc\_16s\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.5 Npp16sc\* nppiMalloc\_16sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

1 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.6 Npp16sc\* nppiMalloc\_16sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.7 Npp16sc\* nppiMalloc\_16sc\_C3 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

3 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.8 Npp16sc\* nppiMalloc\_16sc\_C4 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

4 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.9 Npp16u\* nppiMalloc\_16u\_C1 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.10 Npp16u\* nppiMalloc\_16u\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.11 Npp16u\* nppiMalloc\_16u\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.12 Npp16u\* nppiMalloc\_16u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.13 Npp32f\* nppiMalloc\_32f\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.14 Npp32f\* nppiMalloc\_32f\_C2 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

2 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.15 Npp32f\* nppiMalloc\_32f\_C3 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

3 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.16 Npp32f\* nppiMalloc\_32f\_C4 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

4 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.17 Npp32fc\* nppiMalloc\_32fc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.18 Npp32fc\* nppiMalloc\_32fc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.19 Npp32fc\* nppiMalloc\_32fc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.20 Npp32fc\* nppiMalloc\_32fc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.21 Npp32s\* nppiMalloc\_32s\_C1 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.22 Npp32s\* nppiMalloc\_32s\_C3 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

3 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.23 Npp32s\* nppiMalloc\_32s\_C4 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

4 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.24 Npp32sc\* nppiMalloc\_32sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.25 Npp32sc\* nppiMalloc\_32sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.26 Npp32sc\* nppiMalloc\_32sc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.  
*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.27 Npp32sc\* nppiMalloc\_32sc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.  
*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.28 Npp8u\* nppiMalloc\_8u\_C1 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.29 Npp8u\* nppiMalloc\_8u\_C2 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

2 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.30 Npp8u\* nppiMalloc\_8u\_C3 (int nWidthPixels, int nHeightPixels, int \* pStepBytes)**

3 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.88.2.31 Npp8u\* nppiMalloc\_8u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

## 7.89 Threshold and Compare Operations

Methods for pixel-wise threshold and compare operations.

### Modules

- [Threshold Operations](#)

*Threshold image pixels.*

- [Compare Operations](#)

*Compare the pixels of two images and create a binary result image.*

### 7.89.1 Detailed Description

Methods for pixel-wise threshold and compare operations.

## 7.90 Threshold Operations

Threshold image pixels.

### Functions

- `NppStatus nppiThreshold_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short threshold.*

- **NppStatus** **nppiThreshold\_16u\_C3IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit unsigned short in place threshold.*
- **NppStatus** **nppiThreshold\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short threshold.*
- **NppStatus** **nppiThreshold\_16s\_C3IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short in place threshold.*
- **NppStatus** **nppiThreshold\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 32-bit floating point threshold.*
- **NppStatus** **nppiThreshold\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 32-bit floating point in place threshold.*
- **NppStatus** **nppiThreshold\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_16u\_AC4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- **NppStatus** **nppiThreshold\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GT_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GT_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GT_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GT_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_GT_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GT_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GT_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GT_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GT_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GT_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GT_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GT_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_GT_16s_C3IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`)

*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_GT_32f_C3R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`)

*3 channel 32-bit floating point threshold.*

- `NppStatus nppiThreshold_GT_32f_C3IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`)

*3 channel 32-bit floating point in place threshold.*

- `NppStatus nppiThreshold_GT_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`)

*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`)

*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16u_AC4R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`)

*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16u_AC4IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`)

*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`)

*4 channel 16-bit signed short image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16s_AC4IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`)

*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_32f_AC4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`)

*4 channel 32-bit floating point image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`)

*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_LT_8u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u nThreshold`)

*1 channel 8-bit unsigned char threshold.*

- `NppStatus nppiThreshold_LT_8u_C1R` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u nThreshold`)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_LT_16u_C1R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16u nThreshold`)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_LT_16u_C1IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16u nThreshold`)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_LT_16s_C1R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16s nThreshold`)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_LT_16s_C1IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s nThreshold`)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_LT_32f_C1R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f nThreshold`)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LT_32f_C1IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f nThreshold`)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LT_8u_C3R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`)  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LT_8u_C3IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`)  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_LT_16u_C3R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`)  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_LT_16u_C3IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`)  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_LT_16s_C3R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`)  
*3 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_LT_16s_C3IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`)  
*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LT_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LT_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LT_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_Val_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_Val_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_Val_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_Val_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_Val_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_Val_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_Val_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_Val_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_Val_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_Val_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_Val_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_Val_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_Val_16s_C3IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`, `NppCmpOp eComparisonOperation`)  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_Val_32f_C3R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`, `NppCmpOp eComparisonOperation`)  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_Val_32f_C3IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`, `NppCmpOp eComparisonOperation`)  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_Val_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`, `const Npp8u rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`, `const Npp8u rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16u_AC4R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`, `const Npp16u rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16u_AC4IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`, `const Npp16u rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16s_AC4IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_32f_AC4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`, `NppCmpOp eComparisonOperation`)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GTVal_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_GTVal_16s_C3IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`)  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C3R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`)  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C3IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`)  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GTVal_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`, `const Npp8u rValues[3]`)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u rThresholds[3]`, `const Npp8u rValues[3]`)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_16u_AC4R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`, `const Npp16u rValues[3]`)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_16u_AC4IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16u rThresholds[3]`, `const Npp16u rValues[3]`)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_16s_AC4R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_16s_AC4IR` (`Npp16s *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp16s rThresholds[3]`, `const Npp16s rValues[3]`)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_32f_AC4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVal_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f rThresholds[3]`, `const Npp32f rValues[3]`)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_8u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u nThreshold`, `const Npp8u nValue`)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LTVal_8u_C1IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp8u nThreshold`, `const Npp8u nValue`)

*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

*1 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

*1 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

*1 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

*1 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

*1 channel 32-bit floating point threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

*1 channel 32-bit floating point in place threshold.*

- `NppStatus nppiThreshold_LTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

*3 channel 8-bit unsigned char threshold.*

- `NppStatus nppiThreshold_LTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

*3 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

*3 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

*3 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C3R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp32f rThresholds[3]`, const `Npp32f rValues[3]`)  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LTVal_32f_C3IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp32f rThresholds[3]`, const `Npp32f rValues[3]`)  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LTVal_8u_AC4R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u rThresholds[3]`, const `Npp8u rValues[3]`)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_8u_AC4IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp8u rThresholds[3]`, const `Npp8u rValues[3]`)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16u_AC4R` (const `Npp16u *pSrc`, int `nSrcStep`, `Npp16u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp16u rThresholds[3]`, const `Npp16u rValues[3]`)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16u_AC4IR` (`Npp16u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp16u rThresholds[3]`, const `Npp16u rValues[3]`)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16s_AC4R` (const `Npp16s *pSrc`, int `nSrcStep`, `Npp16s *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp16s rThresholds[3]`, const `Npp16s rValues[3]`)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16s_AC4IR` (`Npp16s *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp16s rThresholds[3]`, const `Npp16s rValues[3]`)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_32f_AC4R` (const `Npp32f *pSrc`, int `nSrcStep`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp32f rThresholds[3]`, const `Npp32f rValues[3]`)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_32f_AC4IR` (`Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp32f rThresholds[3]`, const `Npp32f rValues[3]`)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C1R` (const `Npp8u *pSrc`, int `nSrcStep`, `Npp8u *pDst`, int `nDstStep`, `NppiSize oSizeROI`, const `Npp8u nThresholdLT`, const `Npp8u nValueLT`, const `Npp8u nThresholdGT`, const `Npp8u nValueGT`)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C1IR` (`Npp8u *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`, const `Npp8u nThresholdLT`, const `Npp8u nValueLT`, const `Npp8u nThresholdGT`, const `Npp8u nValueGT`)  
*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)
 

*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)
 

*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)
 

*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)
 

*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)
 

*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LTValGTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)
 

*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])
 

*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])
 

*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])
 

*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])
 

*3 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])  
*3 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_LTValGTVal_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LTValGTVal_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LTValGTVal_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholdsLT[3], const `Npp16s` rValuesLT[3], const `Npp16s` rThresholdsGT[3], const `Npp16s` rValuesGT[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_LTValGTVal_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])  
4 channel 32-bit floating point image threshold, not affecting Alpha.
- `NppStatus nppiThreshold_LTValGTVal_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholdsLT[3], const `Npp32f` rValuesLT[3], const `Npp32f` rThresholdsGT[3], const `Npp32f` rValuesGT[3])  
4 channel 32-bit floating point in place image threshold, not affecting Alpha.

## 7.90.1 Detailed Description

Threshold image pixels.

## 7.90.2 Function Documentation

### 7.90.2.1 `NppStatus nppiThreshold_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

### 7.90.2.2 `NppStatus nppiThreshold_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.3 NppStatus nppiThreshold\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.4 NppStatus nppiThreshold\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.5 NppStatus nppiThreshold\_16s\_C3IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)**

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.6 NppStatus nppiThreshold\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)**

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

### 7.90.2.7 `NppStatus nppiThreshold_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

### 7.90.2.8 `NppStatus nppiThreshold_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

### 7.90.2.9 `NppStatus nppiThreshold_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

### 7.90.2.10 `NppStatus nppiThreshold_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

### 7.90.2.11 `NppStatus nppiThreshold_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.90.2.12 NppStatus nppiThreshold\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.90.2.13 NppStatus nppiThreshold\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.14 NppStatus nppiThreshold\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)**

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.15 NppStatus nppiThreshold\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, NppCmpOp eComparisonOperation)**

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

**7.90.2.16** `NppStatus nppiThreshold_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

**7.90.2.17** `NppStatus nppiThreshold_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

**7.90.2.18** `NppStatus nppiThreshold_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.19** `NppStatus nppiThreshold_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.20** `NppStatus nppiThreshold_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.21** `NppStatus nppiThreshold_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.22** `NppStatus nppiThreshold_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.23** `NppStatus nppiThreshold_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.24** `NppStatus nppiThreshold_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.25** `NppStatus nppiThreshold_GT_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.26** `NppStatus nppiThreshold_GT_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.27 **NppStatus nppiThreshold\_GT\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold)**

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.28 **NppStatus nppiThreshold\_GT\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold)**

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.29** `NppStatus nppiThreshold_GT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.30** `NppStatus nppiThreshold_GT_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.31** `NppStatus nppiThreshold_GT_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.32** `NppStatus nppiThreshold_GT_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.33** `NppStatus nppiThreshold_GT_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.34** `NppStatus nppiThreshold_GT_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.35** `NppStatus nppiThreshold_GT_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.36** `NppStatus nppiThreshold_GT_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.37 NppStatus nppiThreshold\_GT\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.38 NppStatus nppiThreshold\_GT\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.39** `NppStatus nppiThreshold_GT_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.40** `NppStatus nppiThreshold_GT_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.41** `NppStatus nppiThreshold_GT_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.42** `NppStatus nppiThreshold_GT_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.43** `NppStatus nppiThreshold_GT_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.44** `NppStatus nppiThreshold_GT_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.45** `NppStatus nppiThreshold_GT_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.46** `NppStatus nppiThreshold_GT_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.47 NppStatus nppiThreshold\_GT\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.48 NppStatus nppiThreshold\_GT\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.49** `NppStatus nppiThreshold_GTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.50** `NppStatus nppiThreshold_GTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.51** `NppStatus nppiThreshold_GTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.52** `NppStatus nppiThreshold_GTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.53** `NppStatus nppiThreshold_GTVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.54** `NppStatus nppiThreshold_GTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.55** `NppStatus nppiThreshold_GTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.56** `NppStatus nppiThreshold_GTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.57 NppStatus nppiThreshold\_GTVal\_16u\_C1IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.58 NppStatus nppiThreshold\_GTVal\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.59** `NppStatus nppiThreshold_GTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.60** `NppStatus nppiThreshold_GTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.61** `NppStatus nppiThreshold_GTVal_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.62** `NppStatus nppiThreshold_GTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.63** `NppStatus nppiThreshold_GTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.64** `NppStatus nppiThreshold_GTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.65** `NppStatus nppiThreshold_GTVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.66** `NppStatus nppiThreshold_GTVal_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.67** `NppStatus nppiThreshold_GTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.68** `NppStatus nppiThreshold_GTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.69 NppStatus nppiThreshold\_GTVal\_8u\_C1IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)**

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.70 NppStatus nppiThreshold\_GTVal\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)**

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.71 NppStatus nppiThreshold\_GTVal\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])**

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.72 NppStatus nppiThreshold\_GTVal\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])**

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.73** `NppStatus nppiThreshold_LT_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.74** `NppStatus nppiThreshold_LT_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.75** `NppStatus nppiThreshold_LT_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.76** `NppStatus nppiThreshold_LT_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.77** `NppStatus nppiThreshold_LT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.78 NppStatus nppiThreshold\_LT\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])**

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.79 NppStatus nppiThreshold\_LT\_16u\_AC4IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])**

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.80 NppStatus nppiThreshold\_LT\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])**

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.90.2.81 `NppStatus nppiThreshold_LT_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.90.2.82 `NppStatus nppiThreshold_LT_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.83** `NppStatus nppiThreshold_LT_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.84** `NppStatus nppiThreshold_LT_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.85** `NppStatus nppiThreshold_LT_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.86** `NppStatus nppiThreshold_LT_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.87** `NppStatus nppiThreshold_LT_32f_C11R (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.88** `NppStatus nppiThreshold_LT_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.89** `NppStatus nppiThreshold_LT_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrcDst* In-Place Image Pointer.
- nSrcDstStep* In-Place-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.90** `NppStatus nppiThreshold_LT_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.91 NppStatus nppiThreshold\_LT\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.92 NppStatus nppiThreshold\_LT\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.93** `NppStatus nppiThreshold_LT_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.94** `NppStatus nppiThreshold_LT_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.95** `NppStatus nppiThreshold_LT_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.96** `NppStatus nppiThreshold_LT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.97** `NppStatus nppiThreshold_LTVAl_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.98** `NppStatus nppiThreshold_LTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.99** `NppStatus nppiThreshold_LTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.100** `NppStatus nppiThreshold_LTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.101 **NppStatus nppiThreshold\_LTVal\_16s\_C3IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.102 **NppStatus nppiThreshold\_LTVal\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.103 NppStatus nppiThreshold\_LTVal\_16u\_AC4IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])**

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.104 NppStatus nppiThreshold\_LTVal\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])**

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.105** `NppStatus nppiThreshold_LTVal_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.106** `NppStatus nppiThreshold_LTVal_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.107** `NppStatus nppiThreshold_LTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.108** `NppStatus nppiThreshold_LTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.109** `NppStatus nppiThreshold_LTVal_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.110** `NppStatus nppiThreshold_LTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.111** `NppStatus nppiThreshold_LTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.112** `NppStatus nppiThreshold_LTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.113 `NppStatus nppiThreshold_LTVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.90.2.114 `NppStatus nppiThreshold_LTVal_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.115 NppStatus nppiThreshold\_LTVal\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])**

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.116 NppStatus nppiThreshold\_LTVal\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])**

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.117** `NppStatus nppiThreshold_LTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.118** `NppStatus nppiThreshold_LTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.119** `NppStatus nppiThreshold_LTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.120** `NppStatus nppiThreshold_LTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.121** `NppStatus nppiThreshold_LTValGTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.122** `NppStatus nppiThreshold_LTValGTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.123** `NppStatus nppiThreshold_LTValGTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.124 NppStatus nppiThreshold\_LTValGTVal\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)**

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.125 NppStatus nppiThreshold\_LTValGTVal\_16s\_C3IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])**

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.126** `NppStatus nppiThreshold_LTValGTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.127** `NppStatus nppiThreshold_LTValGTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.128** `NppStatus nppiThreshold_LTValGTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.129** `NppStatus nppiThreshold_LTValGTVal_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.130** `NppStatus nppiThreshold_LTValGTVal_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.131** `NppStatus nppiThreshold_LTValGTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.132** `NppStatus nppiThreshold_LTValGTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.133** `NppStatus nppiThreshold_LTValGTVal_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.134** `NppStatus nppiThreshold_LTValGTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.135** `NppStatus nppiThreshold_LTValGTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.136** `NppStatus nppiThreshold_LTValGTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.137** `NppStatus nppiThreshold_LTValGTVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.138** `NppStatus nppiThreshold_LTValGTVal_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.139** `NppStatus nppiThreshold_LTValGTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.140** `NppStatus nppiThreshold_LTValGTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.141** `NppStatus nppiThreshold_LTValGTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.142** `NppStatus nppiThreshold_LTValGTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.90.2.143** `NppStatus nppiThreshold_LTValGTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* Destination-Image Pointer.

*nSrcDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.144** `NppStatus nppiThreshold_LTValGTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.90.2.145** `NppStatus nppiThreshold_Val_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.146 NppStatus nppiThreshold\_Val\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)**

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.147 NppStatus nppiThreshold\_Val\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.148** `NppStatus nppiThreshold_Val_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.149** `NppStatus nppiThreshold_Val_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.150** `NppStatus nppiThreshold_Val_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.151** `NppStatus nppiThreshold_Val_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.152** `NppStatus nppiThreshold_Val_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.153** `NppStatus nppiThreshold_Val_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.154** `NppStatus nppiThreshold_Val_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.155** `NppStatus nppiThreshold_Val_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.156** `NppStatus nppiThreshold_Val_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.157** `NppStatus nppiThreshold_Val_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.158 NppStatus nppiThreshold\_Val\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)**

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.159 NppStatus nppiThreshold\_Val\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)**

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.160** `NppStatus nppiThreshold_Val_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.161** `NppStatus nppiThreshold_Val_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.162** `NppStatus nppiThreshold_Val_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.163** `NppStatus nppiThreshold_Val_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.164 NppStatus nppiThreshold\_Val\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)**

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.165 NppStatus nppiThreshold\_Val\_8u\_C1IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)**

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.166** `NppStatus nppiThreshold_Val_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.167** `NppStatus nppiThreshold_Val_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.90.2.168** `NppStatus nppiThreshold_Val_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

## 7.91 Compare Operations

Compare the pixels of two images and create a binary result image.

### Functions

- **NppStatus nppiCompare\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short image compare.*
- **NppStatus nppiCompare\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short image compare.*
- **NppStatus nppiCompare\_16s\_C4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare.*

- [NppStatus nppiCompare\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 16-bit signed short image compare, not affecting Alpha.*
- [NppStatus nppiCompare\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*1 channel 32-bit floating point image compare.*
- [NppStatus nppiCompare\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*3 channel 32-bit floating point image compare.*
- [NppStatus nppiCompare\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 32-bit floating point image compare.*
- [NppStatus nppiCompare\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 32-bit signed floating point compare, not affecting Alpha.*
- [NppStatus nppiCompareC\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) nConstant, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*1 channel 8-bit unsigned char image compare with constant value.*
- [NppStatus nppiCompareC\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pConstants, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*3 channel 8-bit unsigned char image compare with constant value.*
- [NppStatus nppiCompareC\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pConstants, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 8-bit unsigned char image compare with constant value.*
- [NppStatus nppiCompareC\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pConstants, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 8-bit unsigned char image compare, not affecting Alpha.*
- [NppStatus nppiCompareC\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, const [Npp16u](#) nConstant, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*1 channel 16-bit unsigned short image compare with constant value.*
- [NppStatus nppiCompareC\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, const [Npp16u](#) \*pConstants, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*3 channel 16-bit unsigned short image compare with constant value.*
- [NppStatus nppiCompareC\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, const [Npp16u](#) \*pConstants, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppCmpOp](#) eComparisonOperation)  
*4 channel 16-bit unsigned short image compare with constant value.*

- **NppStatus** `nppiCompareC_16u_AC4R` (const **Npp16u** \*pSrc, int nSrcStep, const **Npp16u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare, not affecting Alpha.*
- **NppStatus** `nppiCompareC_16s_C1R` (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short image compare with constant value.*
- **NppStatus** `nppiCompareC_16s_C3R` (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short image compare with constant value.*
- **NppStatus** `nppiCompareC_16s_C4R` (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare with constant value.*
- **NppStatus** `nppiCompareC_16s_AC4R` (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare, not affecting Alpha.*
- **NppStatus** `nppiCompareC_32f_C1R` (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 32-bit floating point image compare with constant value.*
- **NppStatus** `nppiCompareC_32f_C3R` (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 32-bit floating point image compare with constant value.*
- **NppStatus** `nppiCompareC_32f_C4R` (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit floating point image compare with constant value.*
- **NppStatus** `nppiCompareC_32f_AC4R` (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit signed floating point compare, not affecting Alpha.*
- **NppStatus** `nppiCompareEqualEps_32f_C1R` (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*1 channel 32-bit floating point image compare whether two images are equal within epsilon.*
- **NppStatus** `nppiCompareEqualEps_32f_C3R` (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*3 channel 32-bit floating point image compare whether two images are equal within epsilon.*
- **NppStatus** `nppiCompareEqualEps_32f_C4R` (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit floating point image compare whether two images are equal within epsilon.*

- **NppStatus nppiCompareEqualEps\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.*

### 7.91.1 Detailed Description

Compare the pixels of two images and create a binary result image.

In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u\_C1. False is represented by 0, true by NPP\_MAX\_8U.

### 7.91.2 Function Documentation

**7.91.2.1 NppStatus nppiCompare\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

#### Parameters:

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.2 NppStatus nppiCompare\_16s\_C1R (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.3 NppStatus nppiCompare\_16s\_C3R (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

3 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.91.2.4 NppStatus nppiCompare\_16s\_C4R (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.91.2.5 NppStatus nppiCompare\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.6 NppStatus nppiCompare\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

1 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.7 NppStatus nppiCompare\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

3 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.8 NppStatus nppiCompare\_16u\_C4R** (*const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation*)

4 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.9 NppStatus nppiCompare\_32f\_AC4R** (*const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation*)

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.10** `NppStatus nppiCompare_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.11** `NppStatus nppiCompare_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.12** `NppStatus nppiCompare_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.13** `NppStatus nppiCompare_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.14** `NppStatus nppiCompare_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.15** `NppStatus nppiCompare_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.16** `NppStatus nppiCompare_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.17** `NppStatus nppiCompareC_16s_AC4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.18** `NppStatus nppiCompareC_16s_C1R (const Npp16s * pSrc, int nSrcStep, const Npp16s nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*nConstant* constant value.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.91.2.19** `NppStatus nppiCompareC_16s_C3R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.91.2.20** `NppStatus nppiCompareC_16s_C4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.21** `NppStatus nppiCompareC_16u_AC4R (const Npp16u * pSrc, int nSrcStep, const Npp16u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.22** `NppStatus nppiCompareC_16u_C1R (const Npp16u * pSrc, int nSrcStep, const Npp16u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*nConstant* constant value

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.23** `NppStatus nppiCompareC_16u_C3R (const Npp16u * pSrc, int nSrcStep, const Npp16u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.24** `NppStatus nppiCompareC_16u_C4R (const Npp16u * pSrc, int nSrcStep, const Npp16u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.25** `NppStatus nppiCompareC_32f_AC4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pConstants* pointer to a list of constants, one per color channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.91.2.26 `NppStatus nppiCompareC_32f_C1R (const Npp32f * pSrc, int nSrcStep, const Npp32f nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- nConstant* constant value
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.91.2.27 `NppStatus nppiCompareC_32f_C3R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pConstants* pointer to a list of constants, one per color channel.
- pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.28** `NppStatus nppiCompareC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.29** `NppStatus nppiCompareC_8u_AC4R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.30** `NppStatus nppiCompareC_8u_C1R (const Npp8u * pSrc, int nSrcStep, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*nConstant* constant value.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.31** `NppStatus nppiCompareC_8u_C3R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pConstants* pointer to a list of constant values, one per color channel..

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.32** `NppStatus nppiCompareC_8u_C4R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- pConstants* pointer to a list of constants, one per color channel.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.91.2.33 `NppStatus nppiCompareEqualEps_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.91.2.34 `NppStatus nppiCompareEqualEps_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

- pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.35** `NppStatus nppiCompareEqualEps_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

3 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.36** `NppStatus nppiCompareEqualEps_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.237** `NppStatus nppiCompareEqualEpsC_32f_AC4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.238** `NppStatus nppiCompareEqualEpsC_32f_C1R (const Npp32f * pSrc, int nSrcStep, const Npp32f nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*nConstant* constant value

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.39** `NppStatus nppiCompareEqualEpsC_32f_C3R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.40** `NppStatus nppiCompareEqualEpsC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.92 NPP Signal Processing

### Modules

- [Arithmetic and Logical Operations](#)
- [Conversion Functions](#)
- [Filtering Functions](#)

*Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.*

- [Initialization](#)
- [Statistical Functions](#)

*Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.*

- [Memory Management](#)

## 7.93 Arithmetic and Logical Operations

### Modules

- [Arithmetic Operations](#)
- [Logical And Shift Operations](#)

## 7.94 Arithmetic Operations

### Modules

- [AddC](#)  
*Adds a constant value to each sample of a signal.*
- [AddProductC](#)  
*Adds product of a constant and each sample of a source signal to the each sample of destination signal.*
- [MulC](#)  
*Multiplies each sample of a signal by a constant value.*
- [SubC](#)  
*Subtracts a constant from each sample of a signal.*
- [SubCRev](#)  
*Subtracts each sample of a signal from a constant.*
- [DivC](#)  
*Divides each sample of a signal by a constant.*
- [DivCRev](#)  
*Divides a constant by each sample of a signal.*
- [Add](#)  
*Sample by sample addition of two signals.*
- [AddProduct](#)  
*Adds sample by sample product of two signals to the destination signal.*
- [Mul](#)  
*Sample by sample multiplication the samples of two signals.*
- [Sub](#)  
*Sample by sample subtraction of the samples of two signals.*
- [Div](#)  
*Sample by sample division of the samples of two signals.*
- [Div\\_Round](#)  
*Sample by sample division of the samples of two signals with rounding.*
- [Abs](#)  
*Absolute value of each sample of a signal.*
- [Sqr](#)  
*Squares each sample of a signal.*
- [Sqrt](#)

*Square root of each sample of a signal.*

- [Cubrt](#)

*Cube root of each sample of a signal.*

- [Exp](#)

*E raised to the power of each sample of a signal.*

- [Ln](#)

*Natural logarithm of each sample of a signal.*

- [10Log10](#)

*Ten times the decimal logarithm of each sample of a signal.*

- [SumLn](#)

*Sums up the natural logarithm of each sample of a signal.*

- [Arctan](#)

*Inverse tangent of each sample of a signal.*

- [Normalize](#)

*Normalize each sample of a real or complex signal using offset and division operations.*

- [Cauchy, CauchyD, and CauchyDD2](#)

*Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.*

## 7.95 AddC

Adds a constant value to each sample of a signal.

### Functions

- **NppStatus nppsAddC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal add constant, scale, then clamp to saturated value*
- **NppStatus nppsAddC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned charvector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short vector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal add constant and scale.*
- **NppStatus nppsAddC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integersignal add constant and scale.*
- **NppStatus nppsAddC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.*
- **NppStatus nppsAddC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

- `NppStatus nppsAddC_32f_I(Npp32f nValue, Npp32f *pSrcDst, int nLength)`  
32-bit floating point in place signal add constant.
- `NppStatus nppsAddC_32f(const Npp32f *pSrc, Npp32f nValue, Npp32f *pDst, int nLength)`  
32-bit floating point signal add constant.
- `NppStatus nppsAddC_32fc_I(Npp32fc nValue, Npp32fc *pSrcDst, int nLength)`  
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.
- `NppStatus nppsAddC_32fc(const Npp32fc *pSrc, Npp32fc nValue, Npp32fc *pDst, int nLength)`  
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.
- `NppStatus nppsAddC_64f_I(Npp64f nValue, Npp64f *pSrcDst, int nLength)`  
64-bit floating point, in place signal add constant.
- `NppStatus nppsAddC_64f(const Npp64f *pSrc, Npp64f nValue, Npp64f *pDst, int nLength)`  
64-bit floating pointsignal add constant.
- `NppStatus nppsAddC_64fc_I(Npp64fc nValue, Npp64fc *pSrcDst, int nLength)`  
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.
- `NppStatus nppsAddC_64fc(const Npp64fc *pSrc, Npp64fc nValue, Npp64fc *pDst, int nLength)`  
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

### 7.95.1 Detailed Description

Adds a constant value to each sample of a signal.

### 7.95.2 Function Documentation

#### 7.95.2.1 `NppStatus nppsAddC_16s_ISfs(Npp16s nValue, Npp16s *pSrcDst, int nLength, int nScaleFactor)`

16-bit signed short in place signal add constant, scale, then clamp to saturated value.

#### Parameters:

- pSrcDst* In-Place Signal Pointer.
- nValue* Constant value to be added to each vector element
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.2 NppStatus nppsAddC\_16s\_Sfs (const Npp16s \* pSrc, Npp16s nValue, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc* Source Signal Pointer.
- nValue* Constant value to be added to each vector element
- pDst* Destination Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.3 NppStatus nppsAddC\_16sc\_ISfs (Npp16sc nValue, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrcDst* In-Place Signal Pointer.
- nValue* Constant value to be added to each vector element
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.4 NppStatus nppsAddC\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc nValue, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

**Parameters:**

- pSrc* Source Signal Pointer.
- nValue* Constant value to be added to each vector element
- pDst* Destination Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.5 NppStatus nppsAddC\_16u\_ISfs (Npp16u nValue, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.6 NppStatus nppsAddC\_16u\_Sfs (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short vector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.7 NppStatus nppsAddC\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point signal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.8 NppStatus nppsAddC\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal add constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.9 NppStatus nppsAddC\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.10 NppStatus nppsAddC\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.11 NppStatus nppsAddC\_32s\_ISfs (Npp32s nValue, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal add constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.12 NppStatus nppsAddC\_32s\_Sfs (const Npp32s \* pSrc, Npp32s nValue, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integersignal add constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.13 NppStatus nppsAddC\_32sc\_ISfs (Npp32sc nValue, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.14** `NppStatus nppsAddC_32sc_Sfs (const Npp32sc * pSrc, Npp32sc nValue, Npp32sc * pDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.15** `NppStatus nppsAddC_64f (const Npp64f * pSrc, Npp64f nValue, Npp64f * pDst, int nLength)`

64-bit floating point signal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.16** `NppStatus nppsAddC_64f_I (Npp64f nValue, Npp64f * pSrcDst, int nLength)`

64-bit floating point, in place signal add constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.17 NppStatus nppsAddC\_64fc (const Npp64fc \* pSrc, Npp64fc nValue, Npp64fc \* pDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.18 NppStatus nppsAddC\_64fc\_I (Npp64fc nValue, Npp64fc \* pSrcDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.19 NppStatus nppsAddC\_8u\_ISfs (Npp8u nValue, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal add constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.95.2.20** `NppStatus nppsAddC_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned charvector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.96 AddProductC

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

### Functions

- `NppStatus nppsAddProductC_32f` (const `Npp32f` \*pSrc, `Npp32f` nValue, `Npp32f` \*pDst, int nLength)

*32-bit floating point signal add product of signal times constant to destination signal.*

### 7.96.1 Detailed Description

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

### 7.96.2 Function Documentation

#### 7.96.2.1 `NppStatus nppsAddProductC_32f` (const `Npp32f` \*pSrc, `Npp32f` nValue, `Npp32f` \*pDst, int nLength)

32-bit floating point signal add product of signal times constant to destination signal.

#### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

## 7.97 MulC

Multiplies each sample of a signal by a constant value.

### Functions

- **NppStatus nppsMulC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal times constant, scale, then clamp to saturated value*
- **NppStatus nppsMulC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times constant and scale.*
- **NppStatus nppsMulC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.*

- **NppStatus nppsMulC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal times constant.*
- **NppStatus nppsMulC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal times constant.*
- **NppStatus nppsMulC\_Low\_32f16s** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp16s** \*pDst, int nLength)  
*32-bit floating point signal times constant with output converted to 16-bit signed integer.*
- **NppStatus nppsMulC\_32f16s\_Sfs** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.*
- **NppStatus nppsMulC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.*
- **NppStatus nppsMulC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.*
- **NppStatus nppsMulC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point, in place signal times constant.*
- **NppStatus nppsMulC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal times constant.*
- **NppStatus nppsMulC\_64f64s\_ISfs** (**Npp64f** nValue, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
*64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.*
- **NppStatus nppsMulC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.*
- **NppStatus nppsMulC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.*

### 7.97.1 Detailed Description

Multiplies each sample of a signal by a constant value.

### 7.97.2 Function Documentation

#### 7.97.2.1 **NppStatus nppsMulC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.2 NppStatus nppsMulC\_16s\_Sfs (const Npp16s \* pSrc, Npp16s nValue, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.3 NppStatus nppsMulC\_16sc\_ISfs (Npp16sc nValue, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.97.2.4 **NppStatus nppsMulC\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc nValue, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.97.2.5 **NppStatus nppsMulC\_16u\_ISfs (Npp16u nValue, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.97.2.6 **NppStatus nppsMulC\_16u\_Sfs (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal times constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.7 NppStatus nppsMulC\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point signal times constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.8 NppStatus nppsMulC\_32f16s\_Sfs (const Npp32f \* pSrc, Npp32f nValue, Npp16s \* pDst, int nLength, int nScaleFactor)**

32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nScaleFactor* Integer Result Scaling.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.9 NppStatus nppsMulC\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal times constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.10 NppStatus nppsMulC\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.11 NppStatus nppsMulC\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.12 NppStatus nppsMulC\_32s\_ISfs (Npp32s nValue, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal times constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.13 NppStatus nppsMulC\_32s\_Sfs (const Npp32s \* pSrc, Npp32s nValue, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal times constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.14 NppStatus nppsMulC\_32sc\_ISfs (Npp32sc nValue, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.15 NppStatus nppsMulC\_32sc\_Sfs (const Npp32sc \* pSrc, Npp32sc nValue, Npp32sc \* pDst, int nLength, int nScaleFactor)**

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.16 NppStatus nppsMulC\_64f (const Npp64f \* pSrc, Npp64f nValue, Npp64f \* pDst, int nLength)**

64-bit floating point signal times constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.17 NppStatus nppsMulC\_64f64s\_ISfs (Npp64f nValue, Npp64s \* pDst, int nLength, int nScaleFactor)**

64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.

**Parameters:**

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.18 NppStatus nppsMulC\_64f\_I (Npp64f nValue, Npp64f \* pSrcDst, int nLength)**

64-bit floating point, in place signal times constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.19 NppStatus nppsMulC\_64fc (const Npp64fc \* pSrc, Npp64fc nValue, Npp64fc \* pDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.20 NppStatus nppsMulC\_64fc\_I (Npp64fc nValue, Npp64fc \* pSrcDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.21 NppStatus nppsMulC\_8u\_ISfs (Npp8u nValue, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal times constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.22 NppStatus nppsMulC\_8u\_Sfs (const Npp8u \* pSrc, Npp8u nValue, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.97.2.23 NppStatus nppsMulC\_Low\_32f16s (const Npp32f \* pSrc, Npp32f nValue, Npp16s \* pDst, int nLength)**

32-bit floating point signal times constant with output converted to 16-bit signed integer.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.98 SubC

Subtracts a constant from each sample of a signal.

### Functions

- **NppStatus nppsSubC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value*
- **NppStatus nppsSubC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

- `NppStatus nppsSubC_32f_I` (`Npp32f nValue`, `Npp32f *pSrcDst`, `int nLength`)  
32-bit floating point in place signal subtract constant.
- `NppStatus nppsSubC_32f` (`const Npp32f *pSrc`, `Npp32f nValue`, `Npp32f *pDst`, `int nLength`)  
32-bit floating point signal subtract constant.
- `NppStatus nppsSubC_32fc_I` (`Npp32fc nValue`, `Npp32fc *pSrcDst`, `int nLength`)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.
- `NppStatus nppsSubC_32fc` (`const Npp32fc *pSrc`, `Npp32fc nValue`, `Npp32fc *pDst`, `int nLength`)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.
- `NppStatus nppsSubC_64f_I` (`Npp64f nValue`, `Npp64f *pSrcDst`, `int nLength`)  
64-bit floating point, in place signal subtract constant.
- `NppStatus nppsSubC_64f` (`const Npp64f *pSrc`, `Npp64f nValue`, `Npp64f *pDst`, `int nLength`)  
64-bit floating point signal subtract constant.
- `NppStatus nppsSubC_64fc_I` (`Npp64fc nValue`, `Npp64fc *pSrcDst`, `int nLength`)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.
- `NppStatus nppsSubC_64fc` (`const Npp64fc *pSrc`, `Npp64fc nValue`, `Npp64fc *pDst`, `int nLength`)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

## 7.98.1 Detailed Description

Subtracts a constant from each sample of a signal.

## 7.98.2 Function Documentation

### 7.98.2.1 `NppStatus nppsSubC_16s_ISfs` (`Npp16s nValue`, `Npp16s *pSrcDst`, `int nLength`, `int nScaleFactor`)

16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.2 NppStatus nppsSubC\_16s\_Sfs (const Npp16s \* pSrc, Npp16s nValue, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.3 NppStatus nppsSubC\_16sc\_ISfs (Npp16sc nValue, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.4 NppStatus nppsSubC\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc nValue, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.5 NppStatus nppsSubC\_16u\_ISfs (Npp16u nValue, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.6 NppStatus nppsSubC\_16u\_Sfs (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.7 NppStatus nppsSubC\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.8 NppStatus nppsSubC\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal subtract constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.9 NppStatus nppsSubC\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.10 NppStatus nppsSubC\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.11 NppStatus nppsSubC\_32s\_ISfs (Npp32s nValue, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.12 NppStatus nppsSubC\_32s\_Sfs (const Npp32s \* pSrc, Npp32s nValue, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal subtract constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.13 NppStatus nppsSubC\_32sc\_ISfs (Npp32sc nValue, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.14** `NppStatus nppsSubC_32sc_Sfs (const Npp32sc * pSrc, Npp32sc nValue, Npp32sc * pDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.15** `NppStatus nppsSubC_64f (const Npp64f * pSrc, Npp64f nValue, Npp64f * pDst, int nLength)`

64-bit floating point signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.16** `NppStatus nppsSubC_64f_I (Npp64f nValue, Npp64f * pSrcDst, int nLength)`

64-bit floating point, in place signal subtract constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.17** `NppStatus nppsSubC_64fc (const Npp64fc * pSrc, Npp64fc nValue, Npp64fc * pDst, int nLength)`

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.18** `NppStatus nppsSubC_64fc_I (Npp64fc nValue, Npp64fc * pSrcDst, int nLength)`

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.19** `NppStatus nppsSubC_8u_ISfs (Npp8u nValue, Npp8u * pSrcDst, int nLength, int nScaleFactor)`

8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.98.2.20** `NppStatus nppsSubC_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.99 SubCRev

Subtracts each sample of a signal from a constant.

### Functions

- [NppStatus nppsSubCRev\\_8u\\_ISfs](#) ([Npp8u](#) nValue, [Npp8u](#) \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value*
- [NppStatus nppsSubCRev\\_8u\\_Sfs](#) (const [Npp8u](#) \*pSrc, [Npp8u](#) nValue, [Npp8u](#) \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16u\\_ISfs](#) ([Npp16u](#) nValue, [Npp16u](#) \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16u\\_Sfs](#) (const [Npp16u](#) \*pSrc, [Npp16u](#) nValue, [Npp16u](#) \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16s\\_ISfs](#) ([Npp16s](#) nValue, [Npp16s](#) \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) nValue, [Npp16s](#) \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16sc\\_ISfs](#) ([Npp16sc](#) nValue, [Npp16sc](#) \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_16sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) nValue, [Npp16sc](#) \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- [NppStatus nppsSubCRev\\_32s\\_ISfs](#) ([Npp32s](#) nValue, [Npp32s](#) \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract from constant and scale.*
- [NppStatus nppsSubCRev\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc, [Npp32s](#) nValue, [Npp32s](#) \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integersignal subtract from constant and scale.*
- [NppStatus nppsSubCRev\\_32sc\\_ISfs](#) ([Npp32sc](#) nValue, [Npp32sc](#) \*pSrcDst, int nLength, int nScaleFactor)

*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.*

- **NppStatus nppsSubCRev\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.*
- **NppStatus nppsSubCRev\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract from constant.*
- **NppStatus nppsSubCRev\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.*
- **NppStatus nppsSubCRev\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point, in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract from constant.*
- **NppStatus nppsSubCRev\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.*
- **NppStatus nppsSubCRev\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.*

### 7.99.1 Detailed Description

Subtracts each sample of a signal from a constant.

### 7.99.2 Function Documentation

#### 7.99.2.1 **NppStatus nppsSubCRev\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* **In-Place Signal Pointer.**

*nValue* **Constant value each vector element is to be subtracted from**

*nLength* **Signal Length.**

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.2 NppStatus nppsSubCRev\_16s\_Sfs (const Npp16s \* pSrc, Npp16s nValue, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.3 NppStatus nppsSubCRev\_16sc\_ISfs (Npp16sc nValue, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.4 NppStatus nppsSubCRev\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc nValue, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.99.2.5 NppStatus nppsSubCRev\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.99.2.6 NppStatus nppsSubCRev\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.99.2.7 NppStatus nppsSubCRev\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.8 NppStatus nppsSubCRev\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.9 NppStatus nppsSubCRev\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.10 NppStatus nppsSubCRev\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.11 NppStatus nppsSubCRev\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract from constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.12 NppStatus nppsSubCRev\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integersignal subtract from constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.13 NppStatus nppsSubCRev\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.14** `NppStatus nppsSubCRev_32sc_Sfs (const Npp32sc * pSrc, Npp32sc nValue, Npp32sc * pDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.15** `NppStatus nppsSubCRev_64f (const Npp64f * pSrc, Npp64f nValue, Npp64f * pDst, int nLength)`

64-bit floating point signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.16** `NppStatus nppsSubCRev_64f_I (Npp64f nValue, Npp64f * pSrcDst, int nLength)`

64-bit floating point, in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.17 NppStatus nppsSubCRev\_64fc (const Npp64fc \* pSrc, Npp64fc nValue, Npp64fc \* pDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.18 NppStatus nppsSubCRev\_64fc\_I (Npp64fc nValue, Npp64fc \* pSrcDst, int nLength)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.19 NppStatus nppsSubCRev\_8u\_ISfs (Npp8u nValue, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.99.2.20** `NppStatus nppsSubCRev_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.100 DivC

Divides each sample of a signal by a constant.

### Functions

- **NppStatus nppsDivC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value*
- **NppStatus nppsDivC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal divided by constant.*
- **NppStatus nppsDivC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal divided by constant.*
- **NppStatus nppsDivC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.*
- **NppStatus nppsDivC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.*
- **NppStatus nppsDivC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal divided by constant.*

- `NppStatus nppsDivC_64f` (const `Npp64f *pSrc`, `Npp64f nValue`, `Npp64f *pDst`, int `nLength`)  
*64-bit floating point signal divided by constant.*
- `NppStatus nppsDivC_64fc_I` (`Npp64fc nValue`, `Npp64fc *pSrcDst`, int `nLength`)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.*
- `NppStatus nppsDivC_64fc` (const `Npp64fc *pSrc`, `Npp64fc nValue`, `Npp64fc *pDst`, int `nLength`)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.*

### 7.100.1 Detailed Description

Divides each sample of a signal by a constant.

### 7.100.2 Function Documentation

#### 7.100.2.1 `NppStatus nppsDivC_16s_ISfs` (`Npp16s nValue`, `Npp16s *pSrcDst`, int `nLength`, int `nScaleFactor`)

16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.100.2.2 `NppStatus nppsDivC_16s_Sfs` (const `Npp16s *pSrc`, `Npp16s nValue`, `Npp16s *pDst`, int `nLength`, int `nScaleFactor`)

16-bit signed short signal divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.3 NppStatus nppsDivC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.4 NppStatus nppsDivC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.5 NppStatus nppsDivC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.6 NppStatus nppsDivC\_16u\_Sfs (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.7 NppStatus nppsDivC\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.8 NppStatus nppsDivC\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.9 NppStatus nppsDivC\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.10 NppStatus nppsDivC\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.11 NppStatus nppsDivC\_64f (const Npp64f \* pSrc, Npp64f nValue, Npp64f \* pDst, int nLength)**

64-bit floating point signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.12** `NppStatus nppsDivC_64f_I(Npp64f nValue, Npp64f * pSrcDst, int nLength)`

64-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.13** `NppStatus nppsDivC_64fc(const Npp64fc * pSrc, Npp64fc nValue, Npp64fc * pDst, int nLength)`

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.14** `NppStatus nppsDivC_64fc_I(Npp64fc nValue, Npp64fc * pSrcDst, int nLength)`

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.100.2.15** `NppStatus nppsDivC_8u_ISfs(Npp8u nValue, Npp8u * pSrcDst, int nLength, int nScaleFactor)`

8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.100.2.16 NppStatus nppsDivC\_8u\_Sfs (const Npp8u \* pSrc, Npp8u nValue, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.101 DivCRev

Divides a constant by each sample of a signal.

### Functions

- `NppStatus nppsDivCRev_16u_I(Npp16u nValue, Npp16u *pSrcDst, int nLength)`  
*16-bit unsigned short in place constant divided by signal, then clamp to saturated value.*
- `NppStatus nppsDivCRev_16u(const Npp16u *pSrc, Npp16u nValue, Npp16u *pDst, int nLength)`  
*16-bit unsigned short signal divided by constant, then clamp to saturated value.*
- `NppStatus nppsDivCRev_32f_I(Npp32f nValue, Npp32f *pSrcDst, int nLength)`  
*32-bit floating point in place constant divided by signal.*
- `NppStatus nppsDivCRev_32f(const Npp32f *pSrc, Npp32f nValue, Npp32f *pDst, int nLength)`  
*32-bit floating point constant divided by signal.*

### 7.101.1 Detailed Description

Divides a constant by each sample of a signal.

### 7.101.2 Function Documentation

#### 7.101.2.1 `NppStatus nppsDivCRev_16u(const Npp16u *pSrc, Npp16u nValue, Npp16u *pDst, int nLength)`

16-bit unsigned short signal divided by constant, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.101.2.2 `NppStatus nppsDivCRev_16u_I(Npp16u nValue, Npp16u *pSrcDst, int nLength)`

16-bit unsigned short in place constant divided by signal, then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided by each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.101.2.3 NppStatus nppsDivCRev\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point constant divided by signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.101.2.4 NppStatus nppsDivCRev\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place constant divided by signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be divided by each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.102 Add

Sample by sample addition of two signals.

### Functions

- `NppStatus nppsAdd_16s` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, `Npp16s` \*pDst, int nLength)  
*16-bit signed short signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned int signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_32f` (const `Npp32f` \*pSrc1, const `Npp32f` \*pSrc2, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_64f` (const `Npp64f` \*pSrc1, const `Npp64f` \*pSrc2, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_32fc` (const `Npp32fc` \*pSrc1, const `Npp32fc` \*pSrc2, `Npp32fc` \*pDst, int nLength)  
*32-bit complex floating point signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_64fc` (const `Npp64fc` \*pSrc1, const `Npp64fc` \*pSrc2, `Npp64fc` \*pDst, int nLength)  
*64-bit complex floating point signal add signal, then clamp to saturated value.*
- `NppStatus nppsAdd_8u16u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.*
- `NppStatus nppsAdd_16s32f` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, `Npp32f` \*pDst, int nLength)  
*16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.*
- `NppStatus nppsAdd_8u_Sfs` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_16u_Sfs` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short add signal, scale, then clamp to saturated value.*
- `NppStatus nppsAdd_16s_Sfs` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, `Npp16s` \*pDst, int nLength, int nScaleFactor)

*16-bit signed short add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_32s_Sfs` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, `Npp32s` \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_64s_Sfs` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, `Npp64s` \*pDst, int nLength, int nScaleFactor)

*64-bit signed integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_16sc_Sfs` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, `Npp16sc` \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_32sc_Sfs` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, `Npp32sc` \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)

*16-bit signed short in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)

*32-bit floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_64f_I` (const `Npp64f` \*pSrc, `Npp64f` \*pSrcDst, int nLength)

*64-bit floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_32fc_I` (const `Npp32fc` \*pSrc, `Npp32fc` \*pSrcDst, int nLength)

*32-bit complex floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_64fc_I` (const `Npp64fc` \*pSrc, `Npp64fc` \*pSrcDst, int nLength)

*64-bit complex floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_16s32s_I` (const `Npp16s` \*pSrc, `Npp32s` \*pSrcDst, int nLength)

*16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.*

- `NppStatus nppsAdd_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, int nScaleFactor)

*8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_32s_ISfs` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_16sc_ISfs` (`const Npp16sc *pSrc`, `Npp16sc *pSrcDst`, `int nLength`, `int nScaleFactor`)  
*16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.*
- `NppStatus nppsAdd_32sc_ISfs` (`const Npp32sc *pSrc`, `Npp32sc *pSrcDst`, `int nLength`, `int nScaleFactor`)  
*32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.*

### 7.102.1 Detailed Description

Sample by sample addition of two signals.

### 7.102.2 Function Documentation

#### 7.102.2.1 `NppStatus nppsAdd_16s` (`const Npp16s *pSrc1`, `const Npp16s *pSrc2`, `Npp16s *pDst`, `int nLength`)

16-bit signed short signal add signal, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.102.2.2 `NppStatus nppsAdd_16s32f` (`const Npp16s *pSrc1`, `const Npp16s *pSrc2`, `Npp32f *pDst`, `int nLength`)

16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.3 NppStatus nppsAdd\_16s32s\_I (const Npp16s \* pSrc, Npp32s \* pSrcDst, int nLength)**

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.4 NppStatus nppsAdd\_16s\_I (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.5 NppStatus nppsAdd\_16s\_ISfs (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength, int nScaleFactor)**

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.6** `NppStatus nppsAdd_16s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, Npp16s * pDst, int nLength, int nScaleFactor)`

16-bit signed short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.7** `NppStatus nppsAdd_16sc_ISfs (const Npp16sc * pSrc, Npp16sc * pSrcDst, int nLength, int nScaleFactor)`

16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.8** `NppStatus nppsAdd_16sc_Sfs (const Npp16sc * pSrc1, const Npp16sc * pSrc2, Npp16sc * pDst, int nLength, int nScaleFactor)`

16-bit signed complex short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.9 NppStatus nppsAdd\_16u (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.10 NppStatus nppsAdd\_16u\_ISfs (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.11 NppStatus nppsAdd\_16u\_Sfs (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.12 NppStatus nppsAdd\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)**

32-bit floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.13 NppStatus nppsAdd\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.14 NppStatus nppsAdd\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.15 NppStatus nppsAdd\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.16 NppStatus nppsAdd\_32s\_ISfs (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.17 NppStatus nppsAdd\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.18 NppStatus nppsAdd\_32sc\_ISfs (const Npp32sc \* pSrc, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.19 NppStatus nppsAdd\_32sc\_Sfs (const Npp32sc \* pSrc1, const Npp32sc \* pSrc2, Npp32sc \* pDst, int nLength, int nScaleFactor)**

32-bit signed complex integer add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.20 NppStatus nppsAdd\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, Npp32u \* pDst, int nLength)**

32-bit unsigned int signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.21 NppStatus nppsAdd\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, Npp64f \* pDst, int nLength)**

64-bit floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.22 NppStatus nppsAdd\_64f\_I (const Npp64f \* pSrc, Npp64f \* pSrcDst, int nLength)**

64-bit floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.23 NppStatus nppsAdd\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.24 NppStatus nppsAdd\_64fc\_I (const Npp64fc \* pSrc, Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.102.2.25 NppStatus nppsAdd\_64s\_Sfs (const Npp64s \* pSrc1, const Npp64s \* pSrc2, Npp64s \* pDst, int nLength, int nScaleFactor)**

64-bit signed integer add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.102.2.26 NppStatus nppsAdd\_8u16u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp16u \* pDst, int nLength)**

8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be added to signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.102.2.27 NppStatus nppsAdd\_8u\_ISfs (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.102.2.28 NppStatus nppsAdd\_8u\_Sfs (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.103 AddProduct

Adds sample by sample product of two signals to the destination signal.

### Functions

- **NppStatus** `nppsAddProduct_32f` (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)
 

*32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_64f` (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)
 

*64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_32fc` (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)
 

*32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_64fc` (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)
 

*64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_16s_Sfs` (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)
 

*16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_32s_Sfs` (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)
 

*32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*
- **NppStatus** `nppsAddProduct_16s32s_Sfs` (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)
 

*16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.*

### 7.103.1 Detailed Description

Adds sample by sample product of two signals to the destination signal.

## 7.103.2 Function Documentation

### 7.103.2.1 NppStatus nppsAddProduct\_16s32s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)

16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.103.2.2 NppStatus nppsAddProduct\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)

16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.103.2.3 NppStatus nppsAddProduct\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)

32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.103.2.4** `NppStatus nppsAddProduct_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, Npp32fc * pDst, int nLength)`

32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.103.2.5** `NppStatus nppsAddProduct_32s_Sfs (const Npp32s * pSrc1, const Npp32s * pSrc2, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.103.2.6 NppStatus nppsAddProduct\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, Npp64f \* pDst, int nLength)**

64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.103.2.7 NppStatus nppsAddProduct\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.104 Mul

Sample by sample multiplication the samples of two signals.

### Functions

- **NppStatus nppsMul\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_8u16u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.*
- **NppStatus nppsMul\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_32f32fc** (const **Npp32f** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal time signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_32sc\_Sfs** (const **Npp32sc** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16u16s\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.*

- **NppStatus nppsMul\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.*

- **NppStatus nppsMul\_32s32sc\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.*

- **NppStatus nppsMul\_Low\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16s\_I** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength)

*16-bit signed short in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)

*32-bit floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)

*64-bit complex floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32f32fc\_I** (const **Npp32f** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.*

- **NppStatus nppsMul\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)

*8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsMul\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s32sc\_ISfs** (const **Npp32s** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.*

### 7.104.1 Detailed Description

Sample by sample multiplication the samples of two signals.

### 7.104.2 Function Documentation

#### 7.104.2.1 **NppStatus nppsMul\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)

16-bit signed short signal times signal, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.104.2.2 NppStatus nppsMul\_16s32f (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp32f \* pDst, int nLength)**

16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.3 NppStatus nppsMul\_16s32s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.4 NppStatus nppsMul\_16s\_I (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.5 NppStatus nppsMul\_16s\_ISfs (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength, int nScaleFactor)**

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.6 NppStatus nppsMul\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.7 NppStatus nppsMul\_16sc\_ISfs (const Npp16sc \* pSrc, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.8 NppStatus nppsMul\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.9 NppStatus nppsMul\_16u16s\_Sfs (const Npp16u \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.10 NppStatus nppsMul\_16u\_ISfs (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.11 NppStatus nppsMul\_16u\_Sfs (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal time signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.12 NppStatus nppsMul\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)**

32-bit floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.13 NppStatus nppsMul\_32f32fc (const Npp32f \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.14 NppStatus nppsMul\_32f32fc\_I (const Npp32f \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.15 NppStatus nppsMul\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.16 NppStatus nppsMul\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.104.2.17 NppStatus nppsMul\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)

32-bit complex floating point in place signal times signal, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.104.2.18 NppStatus nppsMul\_32s32sc\_ISfs (const Npp32s \* pSrc, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.104.2.19 NppStatus nppsMul\_32s32sc\_Sfs (const Npp32s \* pSrc1, const Npp32sc \* pSrc2, Npp32sc \* pDst, int nLength, int nScaleFactor)

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.20 NppStatus nppsMul\_32s\_ISfs (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.21 NppStatus nppsMul\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.22 NppStatus nppsMul\_32sc\_ISfs (const Npp32sc \* pSrc, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.23 NppStatus nppsMul\_32sc\_Sfs (const Npp32sc \* pSrc1, const Npp32sc \* pSrc2, Npp32sc \* pDst, int nLength, int nScaleFactor)**

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.24 NppStatus nppsMul\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, Npp64f \* pDst, int nLength)**

64-bit floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.25 NppStatus nppsMul\_64f\_I (const Npp64f \* pSrc, Npp64f \* pSrcDst, int nLength)**

64-bit floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.26 NppStatus nppsMul\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.27 NppStatus nppsMul\_64fc\_I (const Npp64fc \* pSrc, Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.28 NppStatus nppsMul\_8u16u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp16u \* pDst, int nLength)**

8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.29 NppStatus nppsMul\_8u\_ISfs (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.30 NppStatus nppsMul\_8u\_Sfs (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.104.2.31 NppStatus nppsMul\_Low\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.105 Sub

Sample by sample subtraction of the samples of two signals.

### Functions

- **NppStatus nppsSub\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.*
- **NppStatus nppsSub\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_32sc\_Sfs** (const **Npp32sc** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.*

- **NppStatus nppsSub\_16s\_I** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short in place signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point in place signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point in place signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point in place signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsSub\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsSub\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsSub\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsSub\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsSub\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.*

### 7.105.1 Detailed Description

Sample by sample subtraction of the samples of two signals.

### 7.105.2 Function Documentation

#### 7.105.2.1 **NppStatus nppsSub\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)

16-bit signed short signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.2 NppStatus nppsSub\_16s32f (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp32f \* pDst, int nLength)**

16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.3 NppStatus nppsSub\_16s\_I (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.4 NppStatus nppsSub\_16s\_ISfs (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength, int nScaleFactor)**

16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.105.2.5 **NppStatus nppsSub\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.105.2.6 **NppStatus nppsSub\_16sc\_ISfs (const Npp16sc \* pSrc, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.105.2.7 **NppStatus nppsSub\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.8 NppStatus nppsSub\_16u\_ISfs (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.9 NppStatus nppsSub\_16u\_Sfs (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.10 NppStatus nppsSub\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)**

32-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.105.2.11 NppStatus nppsSub\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.105.2.12 NppStatus nppsSub\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.105.2.13 NppStatus nppsSub\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.105.2.14 NppStatus nppsSub\_32s\_ISfs (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.15 NppStatus nppsSub\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.16 NppStatus nppsSub\_32sc\_ISfs (const Npp32sc \* pSrc, Npp32sc \* pSrcDst, int nLength, int nScaleFactor)**

32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.17 NppStatus nppsSub\_32sc\_Sfs (const Npp32sc \* pSrc1, const Npp32sc \* pSrc2, Npp32sc \* pDst, int nLength, int nScaleFactor)**

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.18 NppStatus nppsSub\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, Npp64f \* pDst, int nLength)**

64-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.19 NppStatus nppsSub\_64f\_I (const Npp64f \* pSrc, Npp64f \* pSrcDst, int nLength)**

64-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.20 NppStatus nppsSub\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.21 NppStatus nppsSub\_64fc\_I (const Npp64fc \* pSrc, Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.22 NppStatus nppsSub\_8u\_ISfs (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.105.2.23** `NppStatus nppsSub_8u_Sfs (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.106 Div

Sample by sample division of the samples of two signals.

### Functions

- **NppStatus nppsDiv\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_32s16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.*
- **NppStatus nppsDiv\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)

*32-bit floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)

*64-bit complex floating point in place signal divide signal, then clamp to saturated value.*

### 7.106.1 Detailed Description

Sample by sample division of the samples of two signals.

### 7.106.2 Function Documentation

#### 7.106.2.1 **NppStatus nppsDiv\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

##### Parameters:

*pSrc* **Source Signal Pointer.**

*pSrcDst* **In-Place Signal Pointer.** signal1 divisor elements to be divided into signal2 dividend elements

*nLength* **Signal Length.**

*nScaleFactor* **Integer Result Scaling.**

##### Returns:

**Signal Data Related Error Codes, Length Related Error Codes.**

**7.106.2.2 NppStatus nppsDiv\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.3 NppStatus nppsDiv\_16sc\_ISfs (const Npp16sc \* pSrc, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.4 NppStatus nppsDiv\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit signed complex short signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.5 NppStatus nppsDiv\_16u\_ISfs (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.6 NppStatus nppsDiv\_16u\_Sfs (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.7 NppStatus nppsDiv\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)**

32-bit floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.8 NppStatus nppsDiv\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.9 NppStatus nppsDiv\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.10 NppStatus nppsDiv\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.11 NppStatus nppsDiv\_32s16s\_Sfs (const Npp16s \* pSrc1, const Npp32s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.106.2.12 NppStatus nppsDiv\_32s\_ISfs (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.106.2.13 NppStatus nppsDiv\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)

32-bit signed integer signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.14 NppStatus nppsDiv\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, Npp64f \* pDst, int nLength)**

64-bit floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.15 NppStatus nppsDiv\_64f\_I (const Npp64f \* pSrc, Npp64f \* pSrcDst, int nLength)**

64-bit floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.16 NppStatus nppsDiv\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.17 NppStatus nppsDiv\_64fc\_I (const Npp64fc \* pSrc, Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.18 NppStatus nppsDiv\_8u\_ISfs (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.106.2.19 NppStatus nppsDiv\_8u\_Sfs (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.107 Div\_Round

Sample by sample division of the samples of two signals with rounding.

### Functions

- `NppStatus nppsDiv_Round_8u_Sfs` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*
- `NppStatus nppsDiv_Round_16u_Sfs` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.*
- `NppStatus nppsDiv_Round_16s_Sfs` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, `Npp16s` \*pDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*16-bit signed short signal divide signal, scale, round, then clamp to saturated value.*
- `NppStatus nppsDiv_Round_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.*
- `NppStatus nppsDiv_Round_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.*
- `NppStatus nppsDiv_Round_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)  
*16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.*

### 7.107.1 Detailed Description

Sample by sample division of the samples of two signals with rounding.

### 7.107.2 Function Documentation

#### 7.107.2.1 `NppStatus nppsDiv_Round_16s_ISfs` (const `Npp16s` \* pSrc, `Npp16s` \* pSrcDst, int nLength, `NppRoundMode` nRndMode, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

#### Parameters:

*pSrc* **Source Signal Pointer.**

*pSrcDst* **In-Place Signal Pointer.** signal1 divisor elements to be divided into signal2 dividend elements  
*nLength* **Signal Length.**

*nRndMode* various rounding modes.

*nScaleFactor* **Integer Result Scaling.**

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.107.2.2** `NppStatus nppsDiv_Round_16s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, Npp16s * pDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit signed short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.107.2.3** `NppStatus nppsDiv_Round_16u_ISfs (const Npp16u * pSrc, Npp16u * pSrcDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.107.2.4** `NppStatus nppsDiv_Round_16u_Sfs (const Npp16u * pSrc1, const Npp16u * pSrc2, Npp16u * pDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* Signal Length.

*nRndMode* various rounding modes.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.107.2.5** `NppStatus nppsDiv_Round_8u_ISfs (const Npp8u * pSrc, Npp8u * pSrcDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nRndMode* various rounding modes.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.107.2.6** `NppStatus nppsDiv_Round_8u_Sfs (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nRndMode* various rounding modes.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.108 Abs

Absolute value of each sample of a signal.

### Functions

- `NppStatus nppsAbs_16s` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength)  
*16-bit signed short signal absolute value.*
- `NppStatus nppsAbs_32s` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength)  
*32-bit signed integer signal absolute value.*
- `NppStatus nppsAbs_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal absolute value.*
- `NppStatus nppsAbs_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal absolute value.*
- `NppStatus nppsAbs_16s_I` (`Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short signal absolute value.*
- `NppStatus nppsAbs_32s_I` (`Npp32s` \*pSrcDst, int nLength)  
*32-bit signed integer signal absolute value.*
- `NppStatus nppsAbs_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal absolute value.*
- `NppStatus nppsAbs_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal absolute value.*

### 7.108.1 Detailed Description

Absolute value of each sample of a signal.

### 7.108.2 Function Documentation

#### 7.108.2.1 `NppStatus nppsAbs_16s` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength)

16-bit signed short signal absolute value.

#### Parameters:

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.2 NppStatus nppsAbs\_16s\_I (Npp16s \* pSrcDst, int nLength)**

16-bit signed short signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.3 NppStatus nppsAbs\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.4 NppStatus nppsAbs\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.5 NppStatus nppsAbs\_32s (const Npp32s \* pSrc, Npp32s \* pDst, int nLength)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.6 NppStatus nppsAbs\_32s\_I (Npp32s \* pSrcDst, int nLength)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.7 NppStatus nppsAbs\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.108.2.8 NppStatus nppsAbs\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.109 Sqr

Squares each sample of a signal.

### Functions

- **NppStatus nppsSqr\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal squared.*
- **NppStatus nppsSqr\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal squared.*
- **NppStatus nppsSqr\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*

- `NppStatus nppsSqr_16s_ISfs` (`Npp16s *pSrcDst`, `int nLength`, `int nScaleFactor`)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- `NppStatus nppsSqr_16sc_ISfs` (`Npp16sc *pSrcDst`, `int nLength`, `int nScaleFactor`)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*

### 7.109.1 Detailed Description

Squares each sample of a signal.

### 7.109.2 Function Documentation

#### 7.109.2.1 `NppStatus nppsSqr_16s_ISfs` (`Npp16s *pSrcDst`, `int nLength`, `int nScaleFactor`)

16-bit signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

- pSrcDst* [In-Place Signal Pointer](#).
- nLength* [Signal Length](#).
- nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.109.2.2 `NppStatus nppsSqr_16s_Sfs` (`const Npp16s *pSrc`, `Npp16s *pDst`, `int nLength`, `int nScaleFactor`)

16-bit signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

- pSrc* [Source Signal Pointer](#).
- pDst* [Destination Signal Pointer](#).
- nLength* [Signal Length](#).
- nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.109.2.3 `NppStatus nppsSqr_16sc_ISfs` (`Npp16sc *pSrcDst`, `int nLength`, `int nScaleFactor`)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

- pSrcDst* [In-Place Signal Pointer](#).

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.4** `NppStatus nppsSqr_16sc_Sfs (const Npp16sc * pSrc, Npp16sc * pDst, int nLength, int nScaleFactor)`

16-bit complex signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.5** `NppStatus nppsSqr_16u_ISfs (Npp16u * pSrcDst, int nLength, int nScaleFactor)`

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.6** `NppStatus nppsSqr_16u_Sfs (const Npp16u * pSrc, Npp16u * pDst, int nLength, int nScaleFactor)`

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.7 NppStatus nppsSqr\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.8 NppStatus nppsSqr\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.9 NppStatus nppsSqr\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.10 NppStatus nppsSqr\_32fc\_I (Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.11 NppStatus nppsSqr\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.12 NppStatus nppsSqr\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.13 NppStatus nppsSqr\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.14 NppStatus nppsSqr\_64fc\_I (Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.15** `NppStatus nppsSqr_8u_ISfs (Npp8u * pSrcDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.109.2.16** `NppStatus nppsSqr_8u_Sfs (const Npp8u * pSrc, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.110 Sqrt

Square root of each sample of a signal.

### Functions

- `NppStatus nppsSqrt_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal square root.*
- `NppStatus nppsSqrt_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal square root.*
- `NppStatus nppsSqrt_32fc` (const `Npp32fc` \*pSrc, `Npp32fc` \*pDst, int nLength)  
*32-bit complex floating point signal square root.*
- `NppStatus nppsSqrt_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` \*pDst, int nLength)  
*64-bit complex floating point signal square root.*
- `NppStatus nppsSqrt_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal square root.*
- `NppStatus nppsSqrt_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal square root.*
- `NppStatus nppsSqrt_32fc_I` (`Npp32fc` \*pSrcDst, int nLength)  
*32-bit complex floating point signal square root.*
- `NppStatus nppsSqrt_64fc_I` (`Npp64fc` \*pSrcDst, int nLength)  
*64-bit complex floating point signal square root.*
- `NppStatus nppsSqrt_8u_Sfs` (const `Npp8u` \*pSrc, `Npp8u` \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- `NppStatus nppsSqrt_16u_Sfs` (const `Npp16u` \*pSrc, `Npp16u` \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- `NppStatus nppsSqrt_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- `NppStatus nppsSqrt_16sc_Sfs` (const `Npp16sc` \*pSrc, `Npp16sc` \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- `NppStatus nppsSqrt_64s_Sfs` (const `Npp64s` \*pSrc, `Npp64s` \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*
- `NppStatus nppsSqrt_32s16s_Sfs` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*

- **NppStatus nppsSqrt\_64s16s\_Sfs** (const **Npp64s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*
- **NppStatus nppsSqrt\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16sc\_ISfs** (**Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_64s\_ISfs** (**Npp64s** \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*

### 7.110.1 Detailed Description

Square root of each sample of a signal.

### 7.110.2 Function Documentation

#### 7.110.2.1 **NppStatus nppsSqrt\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.110.2.2 **NppStatus nppsSqrt\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.3 NppStatus nppsSqrt\_16sc\_ISfs (Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.4 NppStatus nppsSqrt\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.5 NppStatus nppsSqrt\_16u\_ISfs (Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.6 NppStatus nppsSqrt\_16u\_Sfs (const Npp16u \* pSrc, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.7 NppStatus nppsSqrt\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.8 NppStatus nppsSqrt\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.9 NppStatus nppsSqrt\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.10 NppStatus nppsSqrt\_32fc\_I (Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.11 NppStatus nppsSqrt\_32s16s\_Sfs (const Npp32s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.12 NppStatus nppsSqrt\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.13 NppStatus nppsSqrt\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.14 NppStatus nppsSqrt\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.15 NppStatus nppsSqrt\_64fc\_I (Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.16 NppStatus nppsSqrt\_64s16s\_Sfs (const Npp64s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.17 NppStatus nppsSqrt\_64s\_ISfs (Npp64s \* pSrcDst, int nLength, int nScaleFactor)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.18 NppStatus nppsSqrt\_64s\_Sfs (const Npp64s \* pSrc, Npp64s \* pDst, int nLength, int nScaleFactor)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.19 NppStatus nppsSqrt\_8u\_ISfs (Npp8u \* pSrcDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.110.2.20 NppStatus nppsSqrt\_8u\_Sfs (const Npp8u \* pSrc, Npp8u \* pDst, int nLength, int nScaleFactor)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.111 Cubrt

Cube root of each sample of a signal.

### Functions

- **NppStatus nppsCubrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal cube root.*
- **NppStatus nppsCubrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.*

### 7.111.1 Detailed Description

Cube root of each sample of a signal.

### 7.111.2 Function Documentation

#### 7.111.2.1 **NppStatus nppsCubrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)

32-bit floating point signal cube root.

##### Parameters:

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.111.2.2 **NppStatus nppsCubrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)

32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

##### Parameters:

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

## 7.112 Exp

E raised to the power of each sample of a signal.

### Functions

- `NppStatus nppsExp_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal exponent.*
- `NppStatus nppsExp_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal exponent.*
- `NppStatus nppsExp_32f64f` (const `Npp32f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*32-bit floating point signal exponent with 64-bit floating point result.*
- `NppStatus nppsExp_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal exponent.*
- `NppStatus nppsExp_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal exponent.*
- `NppStatus nppsExp_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- `NppStatus nppsExp_32s_Sfs` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- `NppStatus nppsExp_64s_Sfs` (const `Npp64s` \*pSrc, `Npp64s` \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*
- `NppStatus nppsExp_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- `NppStatus nppsExp_32s_ISfs` (`Npp32s` \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- `NppStatus nppsExp_64s_ISfs` (`Npp64s` \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*

### 7.112.1 Detailed Description

E raised to the power of each sample of a signal.

### 7.112.2 Function Documentation

#### 7.112.2.1 `NppStatus nppsExp_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.2 NppStatus nppsExp\_16s\_Sfs (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.3 NppStatus nppsExp\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.4 NppStatus nppsExp\_32f64f (const Npp32f \* pSrc, Npp64f \* pDst, int nLength)**

32-bit floating point signal exponent with 64-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.5 NppStatus nppsExp\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.6 NppStatus nppsExp\_32s\_ISfs (Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.7 NppStatus nppsExp\_32s\_Sfs (const Npp32s \* pSrc, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.8 NppStatus nppsExp\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.9 NppStatus nppsExp\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.10 NppStatus nppsExp\_64s\_ISfs (Npp64s \* pSrcDst, int nLength, int nScaleFactor)**

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.112.2.11 NppStatus nppsExp\_64s\_Sfs (const Npp64s \* pSrc, Npp64s \* pDst, int nLength, int nScaleFactor)**

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.113 Ln

Natural logarithm of each sample of a signal.

### Functions

- `NppStatus nppsLn_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f32f` (const `Npp64f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*64-bit floating point signal natural logarithm with 32-bit floating point result.*
- `NppStatus nppsLn_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s_Sfs` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s16s_Sfs` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.*
- `NppStatus nppsLn_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s_ISfs` (`Npp32s` \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*

### 7.113.1 Detailed Description

Natural logarithm of each sample of a signal.

### 7.113.2 Function Documentation

#### 7.113.2.1 `NppStatus nppsLn_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.2 NppStatus nppsLn\_16s\_Sfs (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.3 NppStatus nppsLn\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.4 NppStatus nppsLn\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.5 NppStatus nppsLn\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.6 NppStatus nppsLn\_32s\_ISfs (Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

- pSrcDst* In-Place Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.7 NppStatus nppsLn\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.
- nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.8 NppStatus nppsLn\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.9 NppStatus nppsLn\_64f32f (const Npp64f \* pSrc, Npp32f \* pDst, int nLength)**

64-bit floating point signal natural logarithm with 32-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.113.2.10 NppStatus nppsLn\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.114 10Log10

Ten times the decimal logarithm of each sample of a signal.

### Functions

- `NppStatus npps10Log10_32s_Sfs` (const `Npp32s *pSrc`, `Npp32s *pDst`, int `nLength`, int `nScaleFactor`)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*
- `NppStatus npps10Log10_32s_ISfs` (`Npp32s *pSrcDst`, int `nLength`, int `nScaleFactor`)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*

### 7.114.1 Detailed Description

Ten times the decimal logarithm of each sample of a signal.

### 7.114.2 Function Documentation

#### 7.114.2.1 `NppStatus npps10Log10_32s_ISfs` (`Npp32s *pSrcDst`, int `nLength`, int `nScaleFactor`)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

#### Parameters:

- `pSrcDst` [In-Place Signal Pointer](#).
- `nLength` [Signal Length](#).
- `nScaleFactor` [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.114.2.2 `NppStatus npps10Log10_32s_Sfs` (const `Npp32s *pSrc`, `Npp32s *pDst`, int `nLength`, int `nScaleFactor`)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

#### Parameters:

- `pSrc` [Source Signal Pointer](#).
- `pDst` [Destination Signal Pointer](#).
- `nLength` [Signal Length](#).
- `nScaleFactor` [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.115 SumLn

Sums up the natural logarithm of each sample of a signal.

### Functions

- [NppStatus nppsSumLnGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f SumLn.*
- [NppStatus nppsSumLn\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point signal sum natural logarithm.*
- [NppStatus nppsSumLnGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64f SumLn.*
- [NppStatus nppsSumLn\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point signal sum natural logarithm.*
- [NppStatus nppsSumLnGetBufferSize\\_32f64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f64f SumLn.*
- [NppStatus nppsSumLn\\_32f64f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point input, 64-bit floating point output signal sum natural logarithm.*
- [NppStatus nppsSumLnGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s32f SumLn.*
- [NppStatus nppsSumLn\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.*

### 7.115.1 Detailed Description

Sums up the natural logarithm of each sample of a signal.

### 7.115.2 Function Documentation

#### 7.115.2.1 [NppStatus nppsSumLn\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* Signal Length.

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.115.2.2 NppStatus nppsSumLn\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point signal sum natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.115.2.3 NppStatus nppsSumLn\_32f64f (const Npp32f \* pSrc, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point input, 64-bit floating point output signal sum natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.115.2.4 NppStatus nppsSumLn\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

64-bit floating point signal sum natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.115.2.5 NppStatus nppsSumLnGetBufferSize\_16s32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for 16s32f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.115.2.6 NppStatus nppsSumLnGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for 32f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.115.2.7 NppStatus nppsSumLnGetBufferSize\_32f64f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for 32f64f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_32f64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.115.2.8 NppStatus nppsSumLnGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.116 Arctan

Inverse tangent of each sample of a signal.

### Functions

- `NppStatus nppsArctan_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal inverse tangent.*

### 7.116.1 Detailed Description

Inverse tangent of each sample of a signal.

### 7.116.2 Function Documentation

#### 7.116.2.1 `NppStatus nppsArctan_32f` (const `Npp32f` \* pSrc, `Npp32f` \* pDst, int nLength)

32-bit floating point signal inverse tangent.

##### Parameters:

- pSrc* Source Signal Pointer.
- pDst* Destination Signal Pointer.
- nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.116.2.2 `NppStatus nppsArctan_32f_I` (`Npp32f` \* pSrcDst, int nLength)

32-bit floating point signal inverse tangent.

##### Parameters:

- pSrcDst* In-Place Signal Pointer.
- nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.116.2.3 NppStatus nppsArctan\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.116.2.4 NppStatus nppsArctan\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.117 Normalize

Normalize each sample of a real or complex signal using offset and division operations.

### Functions

- `NppStatus nppsNormalize_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength, `Npp32f` vSub, `Npp32f` vDiv)  
*32-bit floating point signal normalize.*
- `NppStatus nppsNormalize_32fc` (const `Npp32fc` \*pSrc, `Npp32fc` \*pDst, int nLength, `Npp32fc` vSub, `Npp32f` vDiv)  
*32-bit complex floating point signal normalize.*
- `NppStatus nppsNormalize_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength, `Npp64f` vSub, `Npp64f` vDiv)  
*64-bit floating point signal normalize.*
- `NppStatus nppsNormalize_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` \*pDst, int nLength, `Npp64fc` vSub, `Npp64f` vDiv)  
*64-bit complex floating point signal normalize.*
- `NppStatus nppsNormalize_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, `Npp16s` vSub, int vDiv, int nScaleFactor)  
*16-bit signed short signal normalize, scale, then clamp to saturated value.*
- `NppStatus nppsNormalize_16sc_Sfs` (const `Npp16sc` \*pSrc, `Npp16sc` \*pDst, int nLength, `Npp16sc` vSub, int vDiv, int nScaleFactor)  
*16-bit complex signed short signal normalize, scale, then clamp to saturated value.*

### 7.117.1 Detailed Description

Normalize each sample of a real or complex signal using offset and division operations.

### 7.117.2 Function Documentation

#### 7.117.2.1 `NppStatus nppsNormalize_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, `Npp16s` vSub, int vDiv, int nScaleFactor)

16-bit signed short signal normalize, scale, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.117.2.2 NppStatus nppsNormalize\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16sc vSub, int vDiv, int nScaleFactor)**

16-bit complex signed short signal normalize, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.117.2.3 NppStatus nppsNormalize\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f vSub, Npp32f vDiv)**

32-bit floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.117.2.4 NppStatus nppsNormalize\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength, Npp32fc vSub, Npp32fc vDiv)**

32-bit complex floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.117.2.5 NppStatus nppsNormalize\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength, Npp64f vSub, Npp64f vDiv)**

64-bit floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.117.2.6 NppStatus nppsNormalize\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64fc vSub, Npp64fc vDiv)**

64-bit complex floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.118 Cauchy, CauchyD, and CauchyDD2

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

### Functions

- `NppStatus nppsCauchy_32f_I (Npp32f *pSrcDst, int nLength, Npp32f nParam)`  
*32-bit floating point signal Cauchy error calculation.*
- `NppStatus nppsCauchyD_32f_I (Npp32f *pSrcDst, int nLength, Npp32f nParam)`  
*32-bit floating point signal Cauchy first derivative.*
- `NppStatus nppsCauchyDD2_32f_I (Npp32f *pSrcDst, Npp32f *pD2FVal, int nLength, Npp32f nParam)`  
*32-bit floating point signal Cauchy first and second derivatives.*

### 7.118.1 Detailed Description

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

### 7.118.2 Function Documentation

#### 7.118.2.1 `NppStatus nppsCauchy_32f_I (Npp32f *pSrcDst, int nLength, Npp32f nParam)`

32-bit floating point signal Cauchy error calculation.

#### Parameters:

- pSrcDst* [In-Place Signal Pointer](#).
- nLength* [Signal Length](#).
- nParam* constant used in Cauchy formula

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.118.2.2 `NppStatus nppsCauchyD_32f_I (Npp32f *pSrcDst, int nLength, Npp32f nParam)`

32-bit floating point signal Cauchy first derivative.

#### Parameters:

- pSrcDst* [In-Place Signal Pointer](#).
- nLength* [Signal Length](#).
- nParam* constant used in Cauchy formula

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.118.2.3 NppStatus nppsCauchyDD2\_32f\_I (Npp32f \* pSrcDst, Npp32f \* pD2FVal, int nLength, Npp32f nParam)**

32-bit floating point signal Cauchy first and second derivatives.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*pD2FVal* [Source Signal Pointer](#). This signal contains the second derivative of the source signal.

*nLength* [Signal Length](#).

*nParam* constant used in Cauchy formula

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.119 Logical And Shift Operations

### Modules

- [AndC](#)  
*Bitwise AND of a constant and each sample of a signal.*
- [And](#)  
*Sample by sample bitwise AND of samples from two signals.*
- [OrC](#)  
*Bitwise OR of a constant and each sample of a signal.*
- [Or](#)  
*Sample by sample bitwise OR of the samples from two signals.*
- [XorC](#)  
*Bitwise XOR of a constant and each sample of a signal.*
- [Xor](#)  
*Sample by sample bitwise XOR of the samples from two signals.*
- [Not](#)  
*Bitwise NOT of each sample of a signal.*
- [LShiftC](#)  
*Left shifts the bits of each sample of a signal by a constant amount.*
- [RShiftC](#)  
*Right shifts the bits of each sample of a signal by a constant amount.*

## 7.120 AndC

Bitwise AND of a constant and each sample of a signal.

### Functions

- `NppStatus nppsAndC_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal and with constant.*
- `NppStatus nppsAndC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal and with constant.*
- `NppStatus nppsAndC_32u` (const `Npp32u` \*pSrc, `Npp32u` nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal and with constant.*
- `NppStatus nppsAndC_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal and with constant.*
- `NppStatus nppsAndC_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal and with constant.*
- `NppStatus nppsAndC_32u_I` (`Npp32u` nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal and with constant.*

### 7.120.1 Detailed Description

Bitwise AND of a constant and each sample of a signal.

### 7.120.2 Function Documentation

#### 7.120.2.1 `NppStatus nppsAndC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal and with constant.

#### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be anded with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.120.2.2 NppStatus nppsAndC\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.120.2.3 NppStatus nppsAndC\_32u (const Npp32u \* pSrc, Npp32u nValue, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal and with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.120.2.4 NppStatus nppsAndC\_32u\_I (Npp32u nValue, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.120.2.5 NppStatus nppsAndC\_8u (const Npp8u \* pSrc, Npp8u nValue, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal and with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.120.2.6 NppStatus nppsAndC\_8u\_I (Npp8u nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal and with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.121 And

Sample by sample bitwise AND of samples from two signals.

### Functions

- `NppStatus nppsAnd_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal and with signal.*
- `NppStatus nppsAnd_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal and with signal.*
- `NppStatus nppsAnd_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal and with signal.*
- `NppStatus nppsAnd_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal and with signal.*
- `NppStatus nppsAnd_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal and with signal.*
- `NppStatus nppsAnd_32u_I` (const `Npp32u` \*pSrc, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal and with signal.*

### 7.121.1 Detailed Description

Sample by sample bitwise AND of samples from two signals.

### 7.121.2 Function Documentation

#### 7.121.2.1 `NppStatus nppsAnd_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal and with signal.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be anded with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.121.2.2 NppStatus nppsAnd\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal and with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be anded with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.121.2.3 NppStatus nppsAnd\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal and with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be anded with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.121.2.4 NppStatus nppsAnd\_32u\_I (const Npp32u \* pSrc, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned integer in place signal and with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be anded with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.121.2.5 NppStatus nppsAnd\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal and with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.121.2.6 NppStatus nppsAnd\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal and with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.122 OrC

Bitwise OR of a constant and each sample of a signal.

### Functions

- `NppStatus nppsOrC_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal or with constant.*
- `NppStatus nppsOrC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal or with constant.*
- `NppStatus nppsOrC_32u` (const `Npp32u` \*pSrc, `Npp32u` nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal or with constant.*
- `NppStatus nppsOrC_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with constant.*
- `NppStatus nppsOrC_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with constant.*
- `NppStatus nppsOrC_32u_I` (`Npp32u` nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal or with constant.*

### 7.122.1 Detailed Description

Bitwise OR of a constant and each sample of a signal.

### 7.122.2 Function Documentation

#### 7.122.2.1 `NppStatus nppsOrC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal or with constant.

#### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.122.2.2 NppStatus nppsOrC\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.122.2.3 NppStatus nppsOrC\_32u (const Npp32u \* pSrc, Npp32u nValue, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.122.2.4 NppStatus nppsOrC\_32u\_I (Npp32u nValue, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.122.2.5 NppStatus nppsOrC\_8u (const Npp8u \* pSrc, Npp8u nValue, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.122.2.6 NppStatus nppsOrC\_8u\_I (Npp8u nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.123 Or

Sample by sample bitwise OR of the samples from two signals.

### Functions

- `NppStatus nppsOr_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal or with signal.*
- `NppStatus nppsOr_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal or with signal.*
- `NppStatus nppsOr_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal or with signal.*
- `NppStatus nppsOr_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with signal.*
- `NppStatus nppsOr_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with signal.*
- `NppStatus nppsOr_32u_I` (const `Npp32u` \*pSrc, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal or with signal.*

### 7.123.1 Detailed Description

Sample by sample bitwise OR of the samples from two signals.

### 7.123.2 Function Documentation

#### 7.123.2.1 `NppStatus nppsOr_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal or with signal.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.123.2.2 NppStatus nppsOr\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.123.2.3 NppStatus nppsOr\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.123.2.4 NppStatus nppsOr\_32u\_I (const Npp32u \* pSrc, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned integer in place signal or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.123.2.5 NppStatus nppsOr\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.123.2.6 NppStatus nppsOr\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.124 XorC

Bitwise XOR of a constant and each sample of a signal.

### Functions

- `NppStatus nppsXorC_8u` (const `Npp8u` \*pSrc, `Npp8u` nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with constant.*
- `NppStatus nppsXorC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with constant.*
- `NppStatus nppsXorC_32u` (const `Npp32u` \*pSrc, `Npp32u` nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with constant.*
- `NppStatus nppsXorC_8u_I` (`Npp8u` nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with constant.*
- `NppStatus nppsXorC_16u_I` (`Npp16u` nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with constant.*
- `NppStatus nppsXorC_32u_I` (`Npp32u` nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal exclusive or with constant.*

### 7.124.1 Detailed Description

Bitwise XOR of a constant and each sample of a signal.

### 7.124.2 Function Documentation

#### 7.124.2.1 `NppStatus nppsXorC_16u` (const `Npp16u` \*pSrc, `Npp16u` nValue, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal exclusive or with constant.

#### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.124.2.2 NppStatus nppsXorC\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.124.2.3 NppStatus nppsXorC\_32u (const Npp32u \* pSrc, Npp32u nValue, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal exclusive or with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.124.2.4 NppStatus nppsXorC\_32u\_I (Npp32u nValue, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.124.2.5 NppStatus nppsXorC\_8u (const Npp8u \* pSrc, Npp8u nValue, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal exclusive or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.124.2.6 NppStatus nppsXorC\_8u\_I (Npp8u nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.125 Xor

Sample by sample bitwise XOR of the samples from two signals.

### Functions

- `NppStatus nppsXor_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with signal.*
- `NppStatus nppsXor_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with signal.*
- `NppStatus nppsXor_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with signal.*
- `NppStatus nppsXor_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with signal.*
- `NppStatus nppsXor_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with signal.*
- `NppStatus nppsXor_32u_I` (const `Npp32u` \*pSrc, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal exclusive or with signal.*

### 7.125.1 Detailed Description

Sample by sample bitwise XOR of the samples from two signals.

### 7.125.2 Function Documentation

#### 7.125.2.1 `NppStatus nppsXor_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, `Npp16u` \*pDst, int nLength)

16-bit unsigned short signal exclusive or with signal.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.125.2.2 NppStatus nppsXor\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal exclusive or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.125.2.3 NppStatus nppsXor\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal exclusive or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.125.2.4 NppStatus nppsXor\_32u\_I (const Npp32u \* pSrc, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned integer in place signal exclusive or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.125.2.5 NppStatus nppsXor\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal exclusive or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.125.2.6 NppStatus nppsXor\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal exclusive or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be exclusive ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.126 Not

Bitwise NOT of each sample of a signal.

### Functions

- [NppStatus nppsNot\\_8u](#) (const [Npp8u](#) \*pSrc, [Npp8u](#) \*pDst, int nLength)  
*8-bit unsigned char not signal.*
- [NppStatus nppsNot\\_16u](#) (const [Npp16u](#) \*pSrc, [Npp16u](#) \*pDst, int nLength)  
*16-bit unsigned short not signal.*
- [NppStatus nppsNot\\_32u](#) (const [Npp32u](#) \*pSrc, [Npp32u](#) \*pDst, int nLength)  
*32-bit unsigned integer not signal.*
- [NppStatus nppsNot\\_8u\\_I](#) ([Npp8u](#) \*pSrcDst, int nLength)  
*8-bit unsigned char in place not signal.*
- [NppStatus nppsNot\\_16u\\_I](#) ([Npp16u](#) \*pSrcDst, int nLength)  
*16-bit unsigned short in place not signal.*
- [NppStatus nppsNot\\_32u\\_I](#) ([Npp32u](#) \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place not signal.*

### 7.126.1 Detailed Description

Bitwise NOT of each sample of a signal.

### 7.126.2 Function Documentation

#### 7.126.2.1 [NppStatus nppsNot\\_16u](#) (const [Npp16u](#) \*pSrc, [Npp16u](#) \*pDst, int nLength)

16-bit unsigned short not signal.

#### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.126.2.2 NppStatus nppsNot\_16u\_I (Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.126.2.3 NppStatus nppsNot\_32u (const Npp32u \* pSrc, Npp32u \* pDst, int nLength)**

32-bit unsigned integer not signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.126.2.4 NppStatus nppsNot\_32u\_I (Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.126.2.5 NppStatus nppsNot\_8u (const Npp8u \* pSrc, Npp8u \* pDst, int nLength)**

8-bit unsigned char not signal.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.126.2.6** `NppStatus nppsNot_8u_I(Npp8u * pSrcDst, int nLength)`

8-bit unsigned char in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.127 LShiftC

Left shifts the bits of each sample of a signal by a constant amount.

### Functions

- `NppStatus nppsLShiftC_8u` (const `Npp8u` \*pSrc, int nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal left shift with constant.*
- `NppStatus nppsLShiftC_16u` (const `Npp16u` \*pSrc, int nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal left shift with constant.*
- `NppStatus nppsLShiftC_16s` (const `Npp16s` \*pSrc, int nValue, `Npp16s` \*pDst, int nLength)  
*16-bit signed short signal left shift with constant.*
- `NppStatus nppsLShiftC_32u` (const `Npp32u` \*pSrc, int nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal left shift with constant.*
- `NppStatus nppsLShiftC_32s` (const `Npp32s` \*pSrc, int nValue, `Npp32s` \*pDst, int nLength)  
*32-bit signed integer signal left shift with constant.*
- `NppStatus nppsLShiftC_8u_I` (int nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal left shift with constant.*
- `NppStatus nppsLShiftC_16u_I` (int nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal left shift with constant.*
- `NppStatus nppsLShiftC_16s_I` (int nValue, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short in place signal left shift with constant.*
- `NppStatus nppsLShiftC_32u_I` (int nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal left shift with constant.*
- `NppStatus nppsLShiftC_32s_I` (int nValue, `Npp32s` \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal left shift with constant.*

### 7.127.1 Detailed Description

Left shifts the bits of each sample of a signal by a constant amount.

### 7.127.2 Function Documentation

#### 7.127.2.1 `NppStatus nppsLShiftC_16s` (const `Npp16s` \*pSrc, int nValue, `Npp16s` \*pDst, int nLength)

16-bit signed short signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to left shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.2 NppStatus nppsLShiftC\_16s\_I (int nValue, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to left shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.3 NppStatus nppsLShiftC\_16u (const Npp16u \* pSrc, int nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to left shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.4 NppStatus nppsLShiftC\_16u\_I (int nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to left shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.5 NppStatus nppsLShiftC\_32s (const Npp32s \* pSrc, int nValue, Npp32s \* pDst, int nLength)**

32-bit signed integer signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.6 NppStatus nppsLShiftC\_32s\_I (int nValue, Npp32s \* pSrcDst, int nLength)**

32-bit signed integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.7 NppStatus nppsLShiftC\_32u (const Npp32u \* pSrc, int nValue, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.8 NppStatus nppsLShiftC\_32u\_I (int nValue, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.9 NppStatus nppsLShiftC\_8u (const Npp8u \* pSrc, int nValue, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.127.2.10 NppStatus nppsLShiftC\_8u\_I (int nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.128 RShiftC

Right shifts the bits of each sample of a signal by a constant amount.

### Functions

- `NppStatus nppsRShiftC_8u` (const `Npp8u` \*pSrc, int nValue, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char signal right shift with constant.*
- `NppStatus nppsRShiftC_16u` (const `Npp16u` \*pSrc, int nValue, `Npp16u` \*pDst, int nLength)  
*16-bit unsigned short signal right shift with constant.*
- `NppStatus nppsRShiftC_16s` (const `Npp16s` \*pSrc, int nValue, `Npp16s` \*pDst, int nLength)  
*16-bit signed short signal right shift with constant.*
- `NppStatus nppsRShiftC_32u` (const `Npp32u` \*pSrc, int nValue, `Npp32u` \*pDst, int nLength)  
*32-bit unsigned integer signal right shift with constant.*
- `NppStatus nppsRShiftC_32s` (const `Npp32s` \*pSrc, int nValue, `Npp32s` \*pDst, int nLength)  
*32-bit signed integer signal right shift with constant.*
- `NppStatus nppsRShiftC_8u_I` (int nValue, `Npp8u` \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal right shift with constant.*
- `NppStatus nppsRShiftC_16u_I` (int nValue, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal right shift with constant.*
- `NppStatus nppsRShiftC_16s_I` (int nValue, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short in place signal right shift with constant.*
- `NppStatus nppsRShiftC_32u_I` (int nValue, `Npp32u` \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal right shift with constant.*
- `NppStatus nppsRShiftC_32s_I` (int nValue, `Npp32s` \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal right shift with constant.*

### 7.128.1 Detailed Description

Right shifts the bits of each sample of a signal by a constant amount.

### 7.128.2 Function Documentation

#### 7.128.2.1 `NppStatus nppsRShiftC_16s` (const `Npp16s` \*pSrc, int nValue, `Npp16s` \*pDst, int nLength)

16-bit signed short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.2 NppStatus nppsRShiftC\_16s\_I (int nValue, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.3 NppStatus nppsRShiftC\_16u (const Npp16u \* pSrc, int nValue, Npp16u \* pDst, int nLength)**

16-bit unsigned short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.4 NppStatus nppsRShiftC\_16u\_I (int nValue, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.5 NppStatus nppsRShiftC\_32s (const Npp32s \* pSrc, int nValue, Npp32s \* pDst, int nLength)**

32-bit signed integer signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.6 NppStatus nppsRShiftC\_32s\_I (int nValue, Npp32s \* pSrcDst, int nLength)**

32-bit signed integer in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.7 NppStatus nppsRShiftC\_32u (const Npp32u \* pSrc, int nValue, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.8 NppStatus nppsRShiftC\_32u\_I (int nValue, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned signed integer in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.9 NppStatus nppsRShiftC\_8u (const Npp8u \* pSrc, int nValue, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.128.2.10 NppStatus nppsRShiftC\_8u\_I (int nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.129 Conversion Functions

### Modules

- [Convert](#)
- [Threshold](#)

## 7.130 Convert

### Convert

Routines for converting the sample-data type of signals.

- `NppStatus nppsConvert_8s16s` (const `Npp8s` \*pSrc, `Npp16s` \*pDst, int nLength)
- `NppStatus nppsConvert_8s32f` (const `Npp8s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_8u32f` (const `Npp8u` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16s8s_Sfs` (const `Npp16s` \*pSrc, `Npp8s` \*pDst, `Npp32u` nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_16s32s` (const `Npp16s` \*pSrc, `Npp32s` \*pDst, int nLength)
- `NppStatus nppsConvert_16s32f` (const `Npp16s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16u32f` (const `Npp16u` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_32s16s` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength)
- `NppStatus nppsConvert_32s32f` (const `Npp32s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_32s64f` (const `Npp32s` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_32f64f` (const `Npp32f` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_64s64f` (const `Npp64s` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_64f32f` (const `Npp64f` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16s32f_Sfs` (const `Npp16s` \*pSrc, `Npp32f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_16s64f_Sfs` (const `Npp16s` \*pSrc, `Npp64f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s16s_Sfs` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s32f_Sfs` (const `Npp32s` \*pSrc, `Npp32f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s64f_Sfs` (const `Npp32s` \*pSrc, `Npp64f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32f8s_Sfs` (const `Npp32f` \*pSrc, `Npp8s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f8u_Sfs` (const `Npp32f` \*pSrc, `Npp8u` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16s_Sfs` (const `Npp32f` \*pSrc, `Npp16s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16u_Sfs` (const `Npp32f` \*pSrc, `Npp16u` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f32s_Sfs` (const `Npp32f` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64s32s_Sfs` (const `Npp64s` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f16s_Sfs` (const `Npp64f` \*pSrc, `Npp16s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f32s_Sfs` (const `Npp64f` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f64s_Sfs` (const `Npp64f` \*pSrc, `Npp64s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)



### 7.130.1 Function Documentation

- 7.130.1.1 `NppStatus nppsConvert_16s32f (const Npp16s * pSrc, Npp32f * pDst, int nLength)`
- 7.130.1.2 `NppStatus nppsConvert_16s32f_Sfs (const Npp16s * pSrc, Npp32f * pDst, int nLength, int nScaleFactor)`
- 7.130.1.3 `NppStatus nppsConvert_16s32s (const Npp16s * pSrc, Npp32s * pDst, int nLength)`
- 7.130.1.4 `NppStatus nppsConvert_16s64f_Sfs (const Npp16s * pSrc, Npp64f * pDst, int nLength, int nScaleFactor)`
- 7.130.1.5 `NppStatus nppsConvert_16s8s_Sfs (const Npp16s * pSrc, Npp8s * pDst, Npp32u nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.6 `NppStatus nppsConvert_16u32f (const Npp16u * pSrc, Npp32f * pDst, int nLength)`
- 7.130.1.7 `NppStatus nppsConvert_32f16s_Sfs (const Npp32f * pSrc, Npp16s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.8 `NppStatus nppsConvert_32f16u_Sfs (const Npp32f * pSrc, Npp16u * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.9 `NppStatus nppsConvert_32f32s_Sfs (const Npp32f * pSrc, Npp32s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.10 `NppStatus nppsConvert_32f64f (const Npp32f * pSrc, Npp64f * pDst, int nLength)`
- 7.130.1.11 `NppStatus nppsConvert_32f8s_Sfs (const Npp32f * pSrc, Npp8s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.12 `NppStatus nppsConvert_32f8u_Sfs (const Npp32f * pSrc, Npp8u * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.13 `NppStatus nppsConvert_32s16s (const Npp32s * pSrc, Npp16s * pDst, int nLength)`
- 7.130.1.14 `NppStatus nppsConvert_32s16s_Sfs (const Npp32s * pSrc, Npp16s * pDst, int nLength, int nScaleFactor)`
- 7.130.1.15 `NppStatus nppsConvert_32s32f (const Npp32s * pSrc, Npp32f * pDst, int nLength)`
- 7.130.1.16 `NppStatus nppsConvert_32s32f_Sfs (const Npp32s * pSrc, Npp32f * pDst, int nLength, int nScaleFactor)`
- 7.130.1.17 `NppStatus nppsConvert_32s64f (const Npp32s * pSrc, Npp64f * pDst, int nLength)`
- 7.130.1.18 `NppStatus nppsConvert_32s64f_Sfs (const Npp32s * pSrc, Npp64f * pDst, int nLength, int nScaleFactor)`
- 7.130.1.19 `NppStatus nppsConvert_64f16s_Sfs (const Npp64f * pSrc, Npp16s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.20 `NppStatus nppsConvert_64f32f (const Npp64f * pSrc, Npp32f * pDst, int nLength)`
- 7.130.1.21 `NppStatus nppsConvert_64f32s_Sfs (const Npp64f * pSrc, Npp32s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.22 `NppStatus nppsConvert_64f64s_Sfs (const Npp64f * pSrc, Npp64s * pDst, int nLength, NppRoundMode eRoundMode, int nScaleFactor)`
- 7.130.1.23 `NppStatus nppsConvert_64s32s_Sfs (const Npp64s * pSrc, Npp32s * pDst, int nLength,`

## 7.131 Threshold

### Threshold Functions

Performs the threshold operation on the samples of a signal by limiting the sample values by a specified constant value.

- [NppStatus nppsThreshold\\_16s](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit signed short signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit in place signed short signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit signed short complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16sc\\_I](#) ([Npp16sc](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit in place signed short complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32f\\_I](#) ([Npp32f](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit in place floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) \*pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit floating point complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32fc\\_I](#) ([Npp32fc](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit in place floating point complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64f\\_I](#) ([Npp64f](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit in place floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64fc](#) (const [Npp64fc](#) \*pSrc, [Npp64fc](#) \*pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit floating point complex number signal threshold with constant level.*

- **NppStatus** `nppsThreshold_64fc_I` (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **NppCmpOp** nRelOp)  
*64-bit in place floating point complex number signal threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_16s` (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_16s_I` (**Npp16s** \*pSrcDst, int nLength, **Npp16s** nLevel)  
*16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_16sc` (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_16sc_I` (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel)  
*16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_32f` (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel)  
*32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_32f_I` (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel)  
*32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_32fc` (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel)  
*32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_32fc_I` (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel)  
*32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_64f` (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel)  
*64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_64f_I` (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel)  
*64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_64fc` (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel)  
*64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_LT_64fc_I` (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel)  
*64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus** `nppsThreshold_GT_16s` (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.*

- [NppStatus nppsThreshold\\_GT\\_16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel)  
*16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel)  
*16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_16sc\\_I](#) ([Npp16sc](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel)  
*16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength, [Npp32f](#) nLevel)  
*32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32f\\_I](#) ([Npp32f](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel)  
*32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) \*pDst, int nLength, [Npp32f](#) nLevel)  
*32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32fc\\_I](#) ([Npp32fc](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel)  
*32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength, [Npp64f](#) nLevel)  
*64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64f\\_I](#) ([Npp64f](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel)  
*64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64fc](#) (const [Npp64fc](#) \*pSrc, [Npp64fc](#) \*pDst, int nLength, [Npp64f](#) nLevel)  
*64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64fc\\_I](#) ([Npp64fc](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel)  
*64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_LTVa16s](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)  
*16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.*
- [NppStatus nppsThreshold\\_LTVa16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)  
*16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.*
- [NppStatus nppsThreshold\\_LTVa16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel, [Npp16sc](#) nValue)  
*16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*

- **NppStatus nppsThreshold\_LTVAl\_16sc\_I** (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_64f\_I** (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVAl\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_GTVAl\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)  
*16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVAl\_16s\_I** (**Npp16s** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)  
*16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVAl\_16sc** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVAl\_16sc\_I** (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*

- **NppStatus nppsThreshold\_GTVVal\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64f\_I** (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*

### 7.131.1 Function Documentation

#### 7.131.1.1 **NppStatus nppsThreshold\_16s** (const **Npp16s** \* pSrc, **Npp16s** \* pDst, int nLength, **Npp16s** nLevel, **NppCmpOp** nRelOp)

16-bit signed short signal threshold with constant level.

##### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.2 NppStatus nppsThreshold\_16s\_I (Npp16s \* pSrcDst, int nLength, Npp16s nLevel, NppCmpOp nRelOp)**

16-bit in place signed short signal threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.3 NppStatus nppsThreshold\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16s nLevel, NppCmpOp nRelOp)**

16-bit signed short complex number signal threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.4 NppStatus nppsThreshold\_16sc\_I (Npp16sc \* pSrcDst, int nLength, Npp16s nLevel, NppCmpOp nRelOp)**

16-bit in place signed short complex number signal threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.5 NppStatus nppsThreshold\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f nLevel, NppCmpOp nRelOp)**

32-bit floating point signal threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.6 NppStatus nppsThreshold\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nLevel, NppCmpOp nRelOp)**

32-bit in place floating point signal threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.7 NppStatus nppsThreshold\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength, Npp32f nLevel, NppCmpOp nRelOp)**

32-bit floating point complex number signal threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.8 NppStatus nppsThreshold\_32fc\_I (Npp32fc \* pSrcDst, int nLength, Npp32f nLevel, NppCmpOp nRelOp)**

32-bit in place floating point complex number signal threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.9 NppStatus nppsThreshold\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength, Npp64f nLevel, NppCmpOp nRelOp)**

64-bit floating point signal threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.10 NppStatus nppsThreshold\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel, NppCmpOp nRelOp)**

64-bit in place floating point signal threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.11 NppStatus nppsThreshold\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel, NppCmpOp nRelOp)**

64-bit floating point complex number signal threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.12 NppStatus nppsThreshold\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel, NppCmpOp nRelOp)**

64-bit in place floating point complex number signal threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.13 NppStatus nppsThreshold\_GT\_16s (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, Npp16s nLevel)**

16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.14 NppStatus nppsThreshold\_GT\_16s\_I (Npp16s \* pSrcDst, int nLength, Npp16s nLevel)**

16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.15 NppStatus nppsThreshold\_GT\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16s nLevel)**

16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.16 NppStatus nppsThreshold\_GT\_16sc\_I (Npp16sc \* pSrcDst, int nLength, Npp16s nLevel)**

16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.17 NppStatus nppsThreshold\_GT\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f nLevel)**

32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.18 NppStatus nppsThreshold\_GT\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nLevel)**

32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.19 NppStatus nppsThreshold\_GT\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength, Npp32f nLevel)**

32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.20 NppStatus nppsThreshold\_GT\_32fc\_I (Npp32fc \* pSrcDst, int nLength, Npp32f nLevel)**

32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.21 NppStatus nppsThreshold\_GT\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength, Npp64f nLevel)**

64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.22 NppStatus nppsThreshold\_GT\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel)**

64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.23 NppStatus nppsThreshold\_GT\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel)**

64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.24 NppStatus nppsThreshold\_GT\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel)**

64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.25 NppStatus nppsThreshold\_GTVal\_16s (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, Npp16s nLevel, Npp16s nValue)**

16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.26 NppStatus nppsThreshold\_GTVal\_16s\_I (Npp16s \* pSrcDst, int nLength, Npp16s nLevel, Npp16s nValue)**

16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.27 NppStatus nppsThreshold\_GTVal\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16s nLevel, Npp16sc nValue)**

16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.28 NppStatus nppsThreshold\_GTVVal\_16sc\_I (Npp16sc \* pSrcDst, int nLength, Npp16s nLevel, Npp16sc nValue)**

16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.29 NppStatus nppsThreshold\_GTVVal\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f nLevel, Npp32f nValue)**

32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.30 NppStatus nppsThreshold\_GTVVal\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nLevel, Npp32f nValue)**

32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.31 NppStatus nppsThreshold\_GTVal\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength, Npp32f nLevel, Npp32fc nValue)**

32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.32 NppStatus nppsThreshold\_GTVal\_32fc\_I (Npp32fc \* pSrcDst, int nLength, Npp32f nLevel, Npp32fc nValue)**

32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.33 NppStatus nppsThreshold\_GTVal\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.34 NppStatus nppsThreshold\_GTVal\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.35 NppStatus nppsThreshold\_GTVal\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64fc nLevel, Npp64fc nValue)**

64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.36 NppStatus nppsThreshold\_GTVal\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64fc nLevel, Npp64fc nValue)**

64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.37 NppStatus nppsThreshold\_LT\_16s (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, Npp16s nLevel)**

16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.38 NppStatus nppsThreshold\_LT\_16s\_I (Npp16s \* pSrcDst, int nLength, Npp16s nLevel)**

16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.39 NppStatus nppsThreshold\_LT\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16s nLevel)**

16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.40 NppStatus nppsThreshold\_LT\_16sc\_I (Npp16sc \* pSrcDst, int nLength, Npp16s nLevel)**

16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.41 NppStatus nppsThreshold\_LT\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f nLevel)**

32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.42 NppStatus nppsThreshold\_LT\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nLevel)**

32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.43 NppStatus nppsThreshold\_LT\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength, Npp32f nLevel)**

32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.44 NppStatus nppsThreshold\_LT\_32fc\_I (Npp32fc \* pSrcDst, int nLength, Npp32f nLevel)**

32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.45 NppStatus nppsThreshold\_LT\_64f (const Npp64f \* pSrc, Npp64f \* pDst, int nLength, Npp64f nLevel)**

64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.46 NppStatus nppsThreshold\_LT\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel)**

64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.47 NppStatus nppsThreshold\_LT\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel)**

64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.48 NppStatus nppsThreshold\_LT\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel)**

64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.49 NppStatus nppsThreshold\_LTVVal\_16s (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, Npp16s nLevel, Npp16s nValue)**

16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.50 NppStatus nppsThreshold\_LTVVal\_16s\_I (Npp16s \* pSrcDst, int nLength, Npp16s nLevel, Npp16s nValue)**

16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.51 NppStatus nppsThreshold\_LTVVal\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, Npp16s nLevel, Npp16sc nValue)**

16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.131.1.52 NppStatus nppsThreshold\_LTVVal\_16sc\_I (Npp16sc \* pSrcDst, int nLength, Npp16s nLevel, Npp16sc nValue)**

16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.53 NppStatus nppsThreshold\_LTVVal\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength, Npp32f nLevel, Npp32f nValue)**

32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.54 NppStatus nppsThreshold\_LTVVal\_32f\_I (Npp32f \* pSrcDst, int nLength, Npp32f nLevel, Npp32f nValue)**

32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.55** **NppStatus nppsThreshold\_LTVVal\_32fc** (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.56** **NppStatus nppsThreshold\_LTVVal\_32fc\_I** (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.57** **NppStatus nppsThreshold\_LTVVal\_64f** (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*, Npp64f *nValue*)

64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.58 NppStatus nppsThreshold\_LTVVal\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.59 NppStatus nppsThreshold\_LTVVal\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.131.1.60 NppStatus nppsThreshold\_LTVVal\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.132 Filtering Functions

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

### Functions

- `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int *\*hpBufferSize*)
- `NppStatus nppsIntegral_32s` (const `Npp32s` *\*pSrc*, `Npp32s` *\*pDst*, int *nLength*, `Npp8u` *\*pDeviceBuffer*)

### 7.132.1 Detailed Description

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

### 7.132.2 Function Documentation

**7.132.2.1** `NppStatus nppsIntegral_32s` (const `Npp32s` *\*pSrc*, `Npp32s` *\*pDst*, int *nLength*, `Npp8u` *\*pDeviceBuffer*)

**7.132.2.2** `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int *\*hpBufferSize*)

## 7.133 Initialization

### Modules

- [Set](#)
- [Zero](#)
- [Copy](#)

## 7.134 Set

### Set

Set methods for 1D vectors of various types.

The copy methods operate on vector data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

- **NppStatus nppsSet\_8u** (Npp8u nValue, Npp8u \*pDst, int nLength)  
*8-bit unsigned char, vector set method.*
- **NppStatus nppsSet\_16s** (Npp16s nValue, Npp16s \*pDst, int nLength)  
*16-bit integer, vector set method.*
- **NppStatus nppsSet\_16sc** (Npp16sc nValue, Npp16sc \*pDst, int nLength)  
*16-bit integer complex, vector set method.*
- **NppStatus nppsSet\_32s** (Npp32s nValue, Npp32s \*pDst, int nLength)  
*32-bit integer, vector set method.*
- **NppStatus nppsSet\_32sc** (Npp32sc nValue, Npp32sc \*pDst, int nLength)  
*32-bit integer complex, vector set method.*
- **NppStatus nppsSet\_32f** (Npp32f nValue, Npp32f \*pDst, int nLength)  
*32-bit float, vector set method.*
- **NppStatus nppsSet\_32fc** (Npp32fc nValue, Npp32fc \*pDst, int nLength)  
*32-bit float complex, vector set method.*
- **NppStatus nppsSet\_64s** (Npp64s nValue, Npp64s \*pDst, int nLength)  
*64-bit long long integer, vector set method.*
- **NppStatus nppsSet\_64sc** (Npp64sc nValue, Npp64sc \*pDst, int nLength)  
*64-bit long long integer complex, vector set method.*
- **NppStatus nppsSet\_64f** (Npp64f nValue, Npp64f \*pDst, int nLength)  
*64-bit double, vector set method.*
- **NppStatus nppsSet\_64fc** (Npp64fc nValue, Npp64fc \*pDst, int nLength)  
*64-bit double complex, vector set method.*

### 7.134.1 Function Documentation

#### 7.134.1.1 NppStatus nppsSet\_16s (Npp16s nValue, Npp16s \*pDst, int nLength)

16-bit integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.2 NppStatus nppsSet\_16sc (Npp16sc nValue, Npp16sc \* pDst, int nLength)**

16-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.3 NppStatus nppsSet\_32f (Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit float, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.4 NppStatus nppsSet\_32fc (Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit float complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.5 NppStatus nppsSet\_32s (Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.6 NppStatus nppsSet\_32sc (Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*)**

32-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.7 NppStatus nppsSet\_64f (Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit double, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.8 NppStatus nppsSet\_64fc (Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit double complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.  
*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.9 NppStatus nppsSet\_64s (Npp64s *nValue*, Npp64s \* *pDst*, int *nLength*)**

64-bit long long integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.10 NppStatus nppsSet\_64sc (Npp64sc *nValue*, Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.134.1.11 NppStatus nppsSet\_8u (Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.135 Zero

### Zero

Set signals to zero.

- [NppStatus nppsZero\\_8u](#) ([Npp8u](#) \*pDst, int nLength)  
*8-bit unsigned char, vector zero method.*
- [NppStatus nppsZero\\_16s](#) ([Npp16s](#) \*pDst, int nLength)  
*16-bit integer, vector zero method.*
- [NppStatus nppsZero\\_16sc](#) ([Npp16sc](#) \*pDst, int nLength)  
*16-bit integer complex, vector zero method.*
- [NppStatus nppsZero\\_32s](#) ([Npp32s](#) \*pDst, int nLength)  
*32-bit integer, vector zero method.*
- [NppStatus nppsZero\\_32sc](#) ([Npp32sc](#) \*pDst, int nLength)  
*32-bit integer complex, vector zero method.*
- [NppStatus nppsZero\\_32f](#) ([Npp32f](#) \*pDst, int nLength)  
*32-bit float, vector zero method.*
- [NppStatus nppsZero\\_32fc](#) ([Npp32fc](#) \*pDst, int nLength)  
*32-bit float complex, vector zero method.*
- [NppStatus nppsZero\\_64s](#) ([Npp64s](#) \*pDst, int nLength)  
*64-bit long long integer, vector zero method.*
- [NppStatus nppsZero\\_64sc](#) ([Npp64sc](#) \*pDst, int nLength)  
*64-bit long long integer complex, vector zero method.*
- [NppStatus nppsZero\\_64f](#) ([Npp64f](#) \*pDst, int nLength)  
*64-bit double, vector zero method.*
- [NppStatus nppsZero\\_64fc](#) ([Npp64fc](#) \*pDst, int nLength)  
*64-bit double complex, vector zero method.*

### 7.135.1 Function Documentation

#### 7.135.1.1 [NppStatus nppsZero\\_16s](#) ([Npp16s](#) \*pDst, int nLength)

16-bit integer, vector zero method.

#### Parameters:

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.2 NppStatus nppsZero\_16sc (Npp16sc \* *pDst*, int *nLength*)**

16-bit integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.3 NppStatus nppsZero\_32f (Npp32f \* *pDst*, int *nLength*)**

32-bit float, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.4 NppStatus nppsZero\_32fc (Npp32fc \* *pDst*, int *nLength*)**

32-bit float complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.5 NppStatus nppsZero\_32s (Npp32s \* *pDst*, int *nLength*)**

32-bit integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.6 NppStatus nppsZero\_32sc (Npp32sc \* pDst, int nLength)**

32-bit integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.7 NppStatus nppsZero\_64f (Npp64f \* pDst, int nLength)**

64-bit double, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.8 NppStatus nppsZero\_64fc (Npp64fc \* pDst, int nLength)**

64-bit double complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.9 NppStatus nppsZero\_64s (Npp64s \* pDst, int nLength)**

64-bit long long integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.10 NppStatus nppsZero\_64sc (Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.135.1.11 NppStatus nppsZero\_8u (Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.136 Copy

### Copy

Copy methods for various type signals.

Copy methods operate on signal data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

- **NppStatus nppsCopy\_8u** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char, vector copy method*
- **NppStatus nppsCopy\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength)  
*16-bit signed short, vector copy method.*
- **NppStatus nppsCopy\_32s** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer, vector copy method.*
- **NppStatus nppsCopy\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit float, vector copy method.*
- **NppStatus nppsCopy\_64s** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int nLength)  
*64-bit signed integer, vector copy method.*
- **NppStatus nppsCopy\_16sc** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength)  
*16-bit complex short, vector copy method.*
- **NppStatus nppsCopy\_32sc** (const **Npp32sc** \*pSrc, **Npp32sc** \*pDst, int nLength)  
*32-bit complex signed integer, vector copy method.*
- **NppStatus nppsCopy\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex float, vector copy method.*
- **NppStatus nppsCopy\_64sc** (const **Npp64sc** \*pSrc, **Npp64sc** \*pDst, int nLength)  
*64-bit complex signed integer, vector copy method.*
- **NppStatus nppsCopy\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex double, vector copy method.*

### 7.136.1 Function Documentation

#### 7.136.1.1 NppStatus nppsCopy\_16s (const Npp16s \* pSrc, Npp16s \* pDst, int nLength)

16-bit signed short, vector copy method.

#### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.2 NppStatus nppsCopy\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength)**

16-bit complex short, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.3 NppStatus nppsCopy\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.4 NppStatus nppsCopy\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength)**

32-bit complex float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.5 NppStatus nppsCopy\_32s (const Npp32s \* pSrc, Npp32s \* pDst, int nLength)**

32-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.6 NppStatus nppsCopy\_32sc (const Npp32sc \* pSrc, Npp32sc \* pDst, int nLength)**

32-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.7 NppStatus nppsCopy\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength)**

64-bit complex double, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.8 NppStatus nppsCopy\_64s (const Npp64s \* pSrc, Npp64s \* pDst, int nLength)**

64-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.9 NppStatus nppsCopy\_64sc (const Npp64sc \* pSrc, Npp64sc \* pDst, int nLength)**

64-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.136.1.10 NppStatus nppsCopy\_8u (const Npp8u \* pSrc, Npp8u \* pDst, int nLength)**

8-bit unsigned char, vector copy method

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.137 Statistical Functions

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

### Modules

- [MinEvery And MaxEvery Functions](#)

*Performs the min or max operation on the samples of a signal.*

- [Sum](#)
- [Maximum](#)
- [Minimum](#)
- [Mean](#)
- [Standard Deviation](#)
- [Mean And Standard Deviation](#)
- [Minimum\\_Maximum](#)
- [Infinity Norm](#)
- [L1 Norm](#)
- [L2 Norm](#)
- [Infinity Norm Diff](#)
- [L1 Norm Diff](#)
- [L2 Norm Diff](#)
- [Dot Product](#)
- [Count In Range](#)
- [Count Zero Crossings](#)

### 7.137.1 Detailed Description

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

## 7.138 MinEvery And MaxEvery Functions

Performs the min or max operation on the samples of a signal.

### Functions

- [NppStatus nppsMinEvery\\_8u\\_I](#) (const [Npp8u](#) \*pSrc, [Npp8u](#) \*pSrcDst, int nLength)  
*8-bit in place min value for each pair of elements.*
- [NppStatus nppsMinEvery\\_16u\\_I](#) (const [Npp16u](#) \*pSrc, [Npp16u](#) \*pSrcDst, int nLength)  
*16-bit unsigned short integer in place min value for each pair of elements.*
- [NppStatus nppsMinEvery\\_16s\\_I](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pSrcDst, int nLength)  
*16-bit signed short integer in place min value for each pair of elements.*
- [NppStatus nppsMinEvery\\_32s\\_I](#) (const [Npp32s](#) \*pSrc, [Npp32s](#) \*pSrcDst, int nLength)  
*32-bit signed integer in place min value for each pair of elements.*
- [NppStatus nppsMinEvery\\_32f\\_I](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pSrcDst, int nLength)  
*32-bit floating point in place min value for each pair of elements.*
- [NppStatus nppsMinEvery\\_64f\\_I](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pSrcDst, int nLength)  
*64-bit floating point in place min value for each pair of elements.*
- [NppStatus nppsMaxEvery\\_8u\\_I](#) (const [Npp8u](#) \*pSrc, [Npp8u](#) \*pSrcDst, int nLength)  
*8-bit in place max value for each pair of elements.*
- [NppStatus nppsMaxEvery\\_16u\\_I](#) (const [Npp16u](#) \*pSrc, [Npp16u](#) \*pSrcDst, int nLength)  
*16-bit unsigned short integer in place max value for each pair of elements.*
- [NppStatus nppsMaxEvery\\_16s\\_I](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pSrcDst, int nLength)  
*16-bit signed short integer in place max value for each pair of elements.*
- [NppStatus nppsMaxEvery\\_32s\\_I](#) (const [Npp32s](#) \*pSrc, [Npp32s](#) \*pSrcDst, int nLength)  
*32-bit signed integer in place max value for each pair of elements.*
- [NppStatus nppsMaxEvery\\_32f\\_I](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pSrcDst, int nLength)  
*32-bit floating point in place max value for each pair of elements.*

### 7.138.1 Detailed Description

Performs the min or max operation on the samples of a signal.

### 7.138.2 Function Documentation

#### 7.138.2.1 [NppStatus nppsMaxEvery\\_16s\\_I](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pSrcDst, int nLength)

16-bit signed short integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.2 NppStatus nppsMaxEvery\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.3 NppStatus nppsMaxEvery\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.4 NppStatus nppsMaxEvery\_32s\_I (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength)**

32-bit signed integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.5 NppStatus nppsMaxEvery\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.6 NppStatus nppsMinEvery\_16s\_I (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength)**

16-bit signed short integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.7 NppStatus nppsMinEvery\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.8 NppStatus nppsMinEvery\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.9 NppStatus nppsMinEvery\_32s\_I (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength)**

32-bit signed integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.10 NppStatus nppsMinEvery\_64f\_I (const Npp64f \* pSrc, Npp64f \* pSrcDst, int nLength)**

64-bit floating point in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.138.2.11 NppStatus nppsMinEvery\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.139 Sum

### Functions

- [NppStatus nppsSumGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32f.*
- [NppStatus nppsSumGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32fc.*
- [NppStatus nppsSumGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_64f.*
- [NppStatus nppsSumGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_64fc.*
- [NppStatus nppsSumGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16s\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16sc\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16sc32sc\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32s\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16s32s\_Sfs.*
- [NppStatus nppsSum\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector sum method*
- [NppStatus nppsSum\\_32fc](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32fc](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector sum method*
- [NppStatus nppsSum\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector sum method*
- [NppStatus nppsSum\\_64fc](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64fc](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double complex vector sum method*
- [NppStatus nppsSum\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pSum, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit short vector sum with integer scaling method*

- **NppStatus nppsSum\_32s\_Sfs** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*32-bit integer vector sum with integer scaling method*
- **NppStatus nppsSum\_16sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp16sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum with integer scaling method*
- **NppStatus nppsSum\_16sc32sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp32sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum (32bit int complex) with integer scaling method*
- **NppStatus nppsSum\_16s32s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit integer vector sum (32bit) with integer scaling method*

## 7.139.1 Function Documentation

### 7.139.1.1 **NppStatus nppsSum\_16s32s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)

16-bit integer vector sum (32bit) with integer scaling method

#### Parameters:

**pSrc** Source Signal Pointer.

**nLength** Signal Length.

**pSum** Pointer to the output result.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

**nScaleFactor** Integer Result Scaling.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.139.1.2 **NppStatus nppsSum\_16s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)

16-bit short vector sum with integer scaling method

#### Parameters:

**pSrc** Source Signal Pointer.

**nLength** Signal Length.

**pSum** Pointer to the output result.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16s\\_Sfs](#) to determine the minimum number of bytes required.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.1.3** `NppStatus nppsSum_16sc32sc_Sfs (const Npp16sc * pSrc, int nLength, Npp32sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum (32bit int complex) with integer scaling method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsSumGetBufferSize\\_16sc32sc\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.1.4** `NppStatus nppsSum_16sc_Sfs (const Npp16sc * pSrc, int nLength, Npp16sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum with integer scaling method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsSumGetBufferSize\\_16sc\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.1.5** `NppStatus nppsSum_32f (const Npp32f * pSrc, int nLength, Npp32f * pSum, Npp8u * pDeviceBuffer)`

32-bit float vector sum method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.1.6 NppStatus nppsSum\_32fc (const Npp32fc \* pSrc, int nLength, Npp32fc \* pSum, Npp8u \* pDeviceBuffer)**

32-bit float complex vector sum method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.1.7 NppStatus nppsSum\_32s\_Sfs (const Npp32s \* pSrc, int nLength, Npp32s \* pSum, int nScaleFactor, Npp8u \* pDeviceBuffer)**

32-bit integer vector sum with integer scaling method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32s\\_Sfs](#) to determine the minimum number of bytes required.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.139.1.8 NppStatus nppsSum\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pSum, Npp8u \* pDeviceBuffer)

64-bit double vector sum method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.139.1.9 NppStatus nppsSum\_64fc (const Npp64fc \* pSrc, int nLength, Npp64fc \* pSum, Npp8u \* pDeviceBuffer)

64-bit double complex vector sum method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.139.1.10 NppStatus nppsSumGetBufferSize\_16s32s\_Sfs (int nLength, int \* hpBufferSize)

Device scratch buffer size (in bytes) for nppsSum\_16s32s\_Sfs.

#### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

**7.139.1.11 NppStatus nppsSumGetBufferSize\_16s\_Sfs (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsSum\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.12 NppStatus nppsSumGetBufferSize\_16sc32sc\_Sfs (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsSum\_16sc32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.13 NppStatus nppsSumGetBufferSize\_16sc\_Sfs (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsSum\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.14 NppStatus nppsSumGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsSum\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.15 NppStatus nppsSumGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.16 NppStatus nppsSumGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.17 NppStatus nppsSumGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.139.1.18 NppStatus nppsSumGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.140 Maximum

### Functions

- [NppStatus nppsMaxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_16s.*
- [NppStatus nppsMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_32s.*
- [NppStatus nppsMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_32f.*
- [NppStatus nppsMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_64f.*
- [NppStatus nppsMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector max method*
- [NppStatus nppsMax\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector max method*
- [NppStatus nppsMax\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector max method*
- [NppStatus nppsMax\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector max method*
- [NppStatus nppsMaxIndxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_16s.*
- [NppStatus nppsMaxIndxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_32s.*
- [NppStatus nppsMaxIndxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_32f.*
- [NppStatus nppsMaxIndxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_64f.*
- [NppStatus nppsMaxIndx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector max index method*
- [NppStatus nppsMaxIndx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector max index method*

- [NppStatus nppsMaxIndx\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit float vector max index method*

- [NppStatus nppsMaxIndx\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*64-bit float vector max index method*

- [NppStatus nppsMaxAbsGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbs\_16s.*

- [NppStatus nppsMaxAbsGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbs\_32s.*

- [NppStatus nppsMaxAbs\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMaxAbs, [Npp8u](#) \*pDeviceBuffer)

*16-bit integer vector max absolute method*

- [NppStatus nppsMaxAbs\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMaxAbs, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector max absolute method*

- [NppStatus nppsMaxAbsIndxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_16s.*

- [NppStatus nppsMaxAbsIndxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_32s.*

- [NppStatus nppsMaxAbsIndx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMaxAbs, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*16-bit integer vector max absolute index method*

- [NppStatus nppsMaxAbsIndx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMaxAbs, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector max absolute index method*

## 7.140.1 Function Documentation

### 7.140.1.1 [NppStatus nppsMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)

16-bit integer vector max method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMaxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.2 NppStatus nppsMax\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMax, Npp8u \* pDeviceBuffer)**

32-bit float vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMaxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.3 NppStatus nppsMax\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMax, Npp8u \* pDeviceBuffer)**

32-bit integer vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMaxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.4 NppStatus nppsMax\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMax, Npp8u \* pDeviceBuffer)**

64-bit float vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* Signal Length.

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.5 NppStatus nppsMaxAbs\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMaxAbs, Npp8u \* pDeviceBuffer)**

16-bit integer vector max absolute method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMaxAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsGetBufferSize\\_16s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.6 NppStatus nppsMaxAbs\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMaxAbs, Npp8u \* pDeviceBuffer)**

32-bit integer vector max absolute method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMaxAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.7 NppStatus nppsMaxAbsGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxAbs\_16s.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.8 NppStatus nppsMaxAbsGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppsMaxAbs\_32s*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.9 NppStatus nppsMaxAbsIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMaxAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

16-bit integer vector max absolute index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMaxAbs* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.10 NppStatus nppsMaxAbsIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMaxAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit integer vector max absolute index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMaxAbs* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsIndxGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.11 NppStatus nppsMaxAbsIndxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.12 NppStatus nppsMaxAbsIndxGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.13 NppStatus nppsMaxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.14 NppStatus nppsMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.15 NppStatus nppsMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.16 NppStatus nppsMaxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.17 NppStatus nppsMaxIndx\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMax*, int \* *pIndx*, Npp8u \* *pDeviceBuffer*)**

16-bit integer vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.18 NppStatus nppsMaxIndx\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMax, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit float vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.19 NppStatus nppsMaxIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMax, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit integer vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.20 NppStatus nppsMaxIndx\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMax, int \* pIndx, Npp8u \* pDeviceBuffer)**

64-bit float vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.140.1.21 NppStatus nppsMaxIndxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.22 NppStatus nppsMaxIndxGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.23 NppStatus nppsMaxIndxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.140.1.24 NppStatus nppsMaxIndxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.141 Minimum

### Functions

- [NppStatus nppsMinGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_16s.*
- [NppStatus nppsMinGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_32s.*
- [NppStatus nppsMinGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_32f.*
- [NppStatus nppsMinGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_64f.*
- [NppStatus nppsMin\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector min method*
- [NppStatus nppsMin\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector min method*
- [NppStatus nppsMin\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector min method*
- [NppStatus nppsMin\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*64-bit integer vector min method*
- [NppStatus nppsMinIndxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_16s.*
- [NppStatus nppsMinIndxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_32s.*
- [NppStatus nppsMinIndxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_32f.*
- [NppStatus nppsMinIndxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_64f.*
- [NppStatus nppsMinIndx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector min index method*
- [NppStatus nppsMinIndx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector min index method*

- **NppStatus nppsMinIndx\_32f** (const **Npp32f** \*pSrc, int nLength, **Npp32f** \*pMin, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit float vector min index method*

- **NppStatus nppsMinIndx\_64f** (const **Npp64f** \*pSrc, int nLength, **Npp64f** \*pMin, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*64-bit float vector min index method*

- **NppStatus nppsMinAbsGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbs\_16s.*

- **NppStatus nppsMinAbsGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbs\_32s.*

- **NppStatus nppsMinAbs\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMinAbs, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector min absolute method*

- **NppStatus nppsMinAbs\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMinAbs, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector min absolute method*

- **NppStatus nppsMinAbsIndxGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbsIndx\_16s.*

- **NppStatus nppsMinAbsIndxGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbsIndx\_32s.*

- **NppStatus nppsMinAbsIndx\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMinAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector min absolute index method*

- **NppStatus nppsMinAbsIndx\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMinAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector min absolute index method*

## 7.141.1 Function Documentation

### 7.141.1.1 **NppStatus nppsMin\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMin, **Npp8u** \*pDeviceBuffer)

16-bit integer vector min method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMinGetBufferSize\\_16s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.2 NppStatus nppsMin\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMin, Npp8u \* pDeviceBuffer)**

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMinGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.3 NppStatus nppsMin\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, Npp8u \* pDeviceBuffer)**

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsMinGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.4 NppStatus nppsMin\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, Npp8u \* pDeviceBuffer)**

64-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* Signal Length.

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.5 NppStatus nppsMinAbs\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMinAbs, Npp8u \* pDeviceBuffer)**

16-bit integer vector min absolute method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMinAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.6 NppStatus nppsMinAbs\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMinAbs, Npp8u \* pDeviceBuffer)**

32-bit integer vector min absolute method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMinAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.7 NppStatus nppsMinAbsGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinAbs\_16s.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.8 NppStatus nppsMinAbsGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppsMinAbs\_32s*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.9 NppStatus nppsMinAbsIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMinAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

16-bit integer vector min absolute index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMinAbs* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsIndxGetBufferSize\\_16s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.10 NppStatus nppsMinAbsIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMinAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit integer vector min absolute index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMinAbs* Pointer to the output result.

*pIdx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsIdxGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.11 NppStatus nppsMinAbsIdxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinAbsIdx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.12 NppStatus nppsMinAbsIdxGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinAbsIdx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.13 NppStatus nppsMinGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMin\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.14 NppStatus nppsMinGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMin\_32f.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.15 NppStatus nppsMinGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMin\_32s.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.16 NppStatus nppsMinGetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMin\_64f.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.17 NppStatus nppsMinIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMin, int \* pIndx, Npp8u \* pDeviceBuffer)**

16-bit integer vector min index method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.18 NppStatus nppsMinIndx\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMin, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit float vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIndxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.19 NppStatus nppsMinIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit integer vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.20 NppStatus nppsMinIndx\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, int \* pIndx, Npp8u \* pDeviceBuffer)**

64-bit float vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIndxGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.1.21 NppStatus nppsMinIndxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.22 NppStatus nppsMinIndxGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinIndx\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.23 NppStatus nppsMinIdxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinIdx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.141.1.24 NppStatus nppsMinIdxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinIdx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.142 Mean

### Functions

- [NppStatus nppsMeanGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32f.*
- [NppStatus nppsMeanGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32fc.*
- [NppStatus nppsMeanGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_64f.*
- [NppStatus nppsMeanGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_64fc.*
- [NppStatus nppsMeanGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_16s\_Sfs.*
- [NppStatus nppsMeanGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32s\_Sfs.*
- [NppStatus nppsMeanGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_16sc\_Sfs.*
- [NppStatus nppsMean\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector mean method*
- [NppStatus nppsMean\\_32fc](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32fc](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector mean method*
- [NppStatus nppsMean\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector mean method*
- [NppStatus nppsMean\\_64fc](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64fc](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double complex vector mean method*
- [NppStatus nppsMean\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit short vector mean with integer scaling method*
- [NppStatus nppsMean\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector mean with integer scaling method*
- [NppStatus nppsMean\\_16sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc, int nLength, [Npp16sc](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

*16-bit short complex vector mean with integer scaling method*

### 7.142.1 Function Documentation

#### 7.142.1.1 `NppStatus nppsMean_16s_Sfs (const Npp16s * pSrc, int nLength, Npp16s * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short vector mean with integer scaling method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_16s\\_Sfs](#) to determine the minimum number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.142.1.2 `NppStatus nppsMean_16sc_Sfs (const Npp16sc * pSrc, int nLength, Npp16sc * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector mean with integer scaling method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_16sc\\_Sfs](#) to determine the minimum number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.142.1.3 `NppStatus nppsMean_32f (const Npp32f * pSrc, int nLength, Npp32f * pMean, Npp8u * pDeviceBuffer)`

32-bit float vector mean method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.1.4** `NppStatus nppsMean_32fc (const Npp32fc * pSrc, int nLength, Npp32fc * pMean, Npp8u * pDeviceBuffer)`

32-bit float complex vector mean method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.1.5** `NppStatus nppsMean_32s_Sfs (const Npp32s * pSrc, int nLength, Npp32s * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

32-bit integer vector mean with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32s\\_Sfs](#) to determine the minimum number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.142.1.6 `NppStatus nppsMean_64f (const Npp64f * pSrc, int nLength, Npp64f * pMean, Npp8u * pDeviceBuffer)`

64-bit double vector mean method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.142.1.7 `NppStatus nppsMean_64fc (const Npp64fc * pSrc, int nLength, Npp64fc * pMean, Npp8u * pDeviceBuffer)`

64-bit double complex vector mean method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.142.1.8 `NppStatus nppsMeanGetBufferSize_16s_Sfs (int nLength, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppsMean_16s_Sfs`.

#### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

**7.142.1.9 NppStatus nppsMeanGetBufferSize\_16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.142.1.10 NppStatus nppsMeanGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.142.1.11 NppStatus nppsMeanGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.142.1.12 NppStatus nppsMeanGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.142.1.13 NppStatus nppsMeanGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.142.1.14 NppStatus nppsMeanGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.143 Standard Deviation

### Functions

- [NppStatus nppsStdDevGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_32f.*
- [NppStatus nppsStdDevGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_64f.*
- [NppStatus nppsStdDevGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_16s32s\_Sfs.*
- [NppStatus nppsStdDevGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_16s\_Sfs.*
- [NppStatus nppsStdDev\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector standard deviation method*
- [NppStatus nppsStdDev\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector standard deviation method*
- [NppStatus nppsStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector standard deviation method (return value is 32-bit)*
- [NppStatus nppsStdDev\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector standard deviation method (return value is also 16-bit)*

### 7.143.1 Function Documentation

#### 7.143.1.1 [NppStatus nppsStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit float vector standard deviation method (return value is 32-bit)

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pStdDev* Pointer to the output result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.143.1.2 NppStatus nppsStdDev\_16s\_Sfs (const Npp16s \* pSrc, int nLength, Npp16s \* pStdDev, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit float vector standard deviation method (return value is also 16-bit)

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pStdDev* Pointer to the output result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.143.1.3 NppStatus nppsStdDev\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pStdDev, Npp8u \* pDeviceBuffer)**

32-bit float vector standard deviation method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pStdDev* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.143.1.4 NppStatus nppsStdDev\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pStdDev, Npp8u \* pDeviceBuffer)**

64-bit float vector standard deviation method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pStdDev* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.143.1.5 NppStatus nppsStdDevGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.143.1.6 NppStatus nppsStdDevGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.143.1.7 NppStatus nppsStdDevGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.143.1.8 NppStatus nppsStdDevGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.144 Mean And Standard Deviation

### Functions

- [NppStatus nppsMeanStdDevGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_32f.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_64f.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s32s\_Sfs.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s\_Sfs.*
- [NppStatus nppsMeanStdDev\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMean, [Npp32f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector mean and standard deviation method*
- [NppStatus nppsMeanStdDev\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector mean and standard deviation method*
- [NppStatus nppsMeanStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector mean and standard deviation method (return values are 32-bit)*
- [NppStatus nppsMeanStdDev\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMean, [Npp16s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector mean and standard deviation method (return values are also 16-bit)*

### 7.144.1 Function Documentation

#### 7.144.1.1 [NppStatus nppsMeanStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit float vector mean and standard deviation method (return values are 32-bit)

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.1.2 NppStatus nppsMeanStdDev\_16s\_Sfs (const Npp16s \* pSrc, int nLength, Npp16s \* pMean, Npp16s \* pStdDev, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit float vector mean and standard deviation method (return values are also 16-bit)

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.1.3 NppStatus nppsMeanStdDev\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMean, Npp32f \* pStdDev, Npp8u \* pDeviceBuffer)**

32-bit float vector mean and standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.1.4 NppStatus nppsMeanStdDev\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMean, Npp64f \* pStdDev, Npp8u \* pDeviceBuffer)**

64-bit float vector mean and standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.1.5** `NppStatus nppsMeanStdDevGetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppsMeanStdDev_16s32s_Sfs`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.144.1.6** `NppStatus nppsMeanStdDevGetBufferSize_16s_Sfs (int nLength, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppsMeanStdDev_16s_Sfs`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.144.1.7** `NppStatus nppsMeanStdDevGetBufferSize_32f (int nLength, int * hpBufferSize)`

Device scratch buffer size (in bytes) for `nppsMeanStdDev_32f`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: `hpBufferSize` is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.144.1.8 NppStatus nppsMeanStdDevGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMeanStdDev\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.145 `Minimum_Maximum`

### Functions

- [NppStatus nppsMinMaxGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_8u.*
- [NppStatus nppsMinMaxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_16s.*
- [NppStatus nppsMinMaxGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_16u.*
- [NppStatus nppsMinMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32s.*
- [NppStatus nppsMinMaxGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32u.*
- [NppStatus nppsMinMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32f.*
- [NppStatus nppsMinMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_64f.*
- [NppStatus nppsMinMax\\_8u](#) (const [Npp8u](#) \*pSrc, int nLength, [Npp8u](#) \*pMin, [Npp8u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*8-bit char vector min and max method*
- [NppStatus nppsMinMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short vector min and max method*
- [NppStatus nppsMinMax\\_16u](#) (const [Npp16u](#) \*pSrc, int nLength, [Npp16u](#) \*pMin, [Npp16u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short vector min and max method*
- [NppStatus nppsMinMax\\_32u](#) (const [Npp32u](#) \*pSrc, int nLength, [Npp32u](#) \*pMin, [Npp32u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned int vector min and max method*
- [NppStatus nppsMinMax\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, [Npp32s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed int vector min and max method*
- [NppStatus nppsMinMax\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, [Npp32f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector min and max method*
- [NppStatus nppsMinMax\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, [Npp64f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)

*64-bit double vector min and max method*

- [NppStatus nppsMinMaxIdxGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_8u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_16s.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_16u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32s.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32f.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_64f.*
- [NppStatus nppsMinMaxIdx\\_8u](#) (const [Npp8u](#) \*pSrc, int nLength, [Npp8u](#) \*pMin, int \*pMinIdx, [Npp8u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*8-bit char vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, int \*pMinIdx, [Npp16s](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_16u](#) (const [Npp16u](#) \*pSrc, int nLength, [Npp16u](#) \*pMin, int \*pMinIdx, [Npp16u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, int \*pMinIdx, [Npp32s](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32u](#) (const [Npp32u](#) \*pSrc, int nLength, [Npp32u](#) \*pMin, int \*pMinIdx, [Npp32u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, int \*pMinIdx, [Npp32f](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, int \*pMinIdx, [Npp64f](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector min and max with indices method*

## 7.145.1 Function Documentation

### 7.145.1.1 `NppStatus nppsMinMax_16s (const Npp16s * pSrc, int nLength, Npp16s * pMin, Npp16s * pMax, Npp8u * pDeviceBuffer)`

16-bit signed short vector min and max method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_16s](#) to determine the minimum number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.145.1.2 `NppStatus nppsMinMax_16u (const Npp16u * pSrc, int nLength, Npp16u * pMin, Npp16u * pMax, Npp8u * pDeviceBuffer)`

16-bit unsigned short vector min and max method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_16u](#) to determine the minimum number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.145.1.3 `NppStatus nppsMinMax_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, Npp32f * pMax, Npp8u * pDeviceBuffer)`

32-bit float vector min and max method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.4 NppStatus nppsMinMax\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, Npp32s \* pMax, Npp8u \* pDeviceBuffer)**

32-bit signed int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.5 NppStatus nppsMinMax\_32u (const Npp32u \* pSrc, int nLength, Npp32u \* pMin, Npp32u \* pMax, Npp8u \* pDeviceBuffer)**

32-bit unsigned int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.6 NppStatus nppsMinMax\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, Npp64f \* pMax, Npp8u \* pDeviceBuffer)**

64-bit double vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.145.1.7 NppStatus nppsMinMax\_8u (const Npp8u \* pSrc, int nLength, Npp8u \* pMin, Npp8u \* pMax, Npp8u \* pDeviceBuffer)

8-bit char vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.145.1.8 NppStatus nppsMinMaxGetBufferSize\_16s (int nLength, int \* hpBufferSize)

Device-buffer size (in bytes) for nppsMinMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

#### 7.145.1.9 NppStatus nppsMinMaxGetBufferSize\_16u (int nLength, int \* hpBufferSize)

Device-buffer size (in bytes) for nppsMinMax\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.10 NppStatus nppsMinMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.11 NppStatus nppsMinMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32s*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.12 NppStatus nppsMinMaxGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32u*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.13 NppStatus nppsMinMaxGetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsMinMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.14 NppStatus nppsMinMaxGetBufferSize\_8u (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsMinMax\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.15 NppStatus nppsMinMaxIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMin, int \* pMinIndx, Npp16s \* pMax, int \* pMaxIndx, Npp8u \* pDeviceBuffer)**

16-bit signed short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_16s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.16 NppStatus nppsMinMaxIndx\_16u (const Npp16u \* pSrc, int nLength, Npp16u \* pMin, int \* pMinIndx, Npp16u \* pMax, int \* pMaxIndx, Npp8u \* pDeviceBuffer)**

16-bit unsigned short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_16u](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.17 NppStatus nppsMinMaxIndx\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMin, int \* pMinIndx, Npp32f \* pMax, int \* pMaxIndx, Npp8u \* pDeviceBuffer)**

32-bit float vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.18 NppStatus nppsMinMaxIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, int \* pMinIndx, Npp32s \* pMax, int \* pMaxIndx, Npp8u \* pDeviceBuffer)**

32-bit signed short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIdx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIdx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIdxGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.19 NppStatus nppsMinMaxIdx\_32u (const Npp32u \* pSrc, int nLength, Npp32u \* pMin, int \* pMinIdx, Npp32u \* pMax, int \* pMaxIdx, Npp8u \* pDeviceBuffer)**

32-bit unsigned short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIdx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIdx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIdxGetBufferSize\\_32u](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.20 NppStatus nppsMinMaxIdx\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, int \* pMinIdx, Npp64f \* pMax, int \* pMaxIdx, Npp8u \* pDeviceBuffer)**

64-bit float vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIdx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIdx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIdxGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.21 NppStatus nppsMinMaxIndx\_8u (const Npp8u \* pSrc, int nLength, Npp8u \* pMin, int \* pMinIdx, Npp8u \* pMax, int \* pMaxIdx, Npp8u \* pDeviceBuffer)**

8-bit char vector min and max with indices method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pMin* Pointer to the min output result.

*pMinIdx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIdx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_8u](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.1.22 NppStatus nppsMinMaxIndxGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsMinMaxIndx\_16s.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.23 NppStatus nppsMinMaxIndxGetBufferSize\_16u (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsMinMaxIndx\_16u.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.24 NppStatus nppsMinMaxIdxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.25 NppStatus nppsMinMaxIdxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.26 NppStatus nppsMinMaxIdxGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.27 NppStatus nppsMinMaxIdxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.145.1.28 NppStatus nppsMinMaxIndxGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIndx\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.146 Infinity Norm

### Functions

- [NppStatus nppsNormInfGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_32f.*
- [NppStatus nppsNorm\\_Inf\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector C norm method*
- [NppStatus nppsNormInfGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_64f.*
- [NppStatus nppsNorm\\_Inf\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector C norm method*
- [NppStatus nppsNormInfGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32f.*
- [NppStatus nppsNorm\\_Inf\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector C norm method, return value is 32-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_32fc32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_32fc32f.*
- [NppStatus nppsNorm\\_Inf\\_32fc32f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector C norm method, return value is 32-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_64fc64f.*
- [NppStatus nppsNorm\\_Inf\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector C norm method, return value is 64-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_Inf\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector C norm method, return value is 32-bit signed integer.*

## 7.146.1 Function Documentation

### 7.146.1.1 `NppStatus nppsNorm_Inf_16s32f (const Npp16s * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit float.

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.146.1.2 `NppStatus nppsNorm_Inf_16s32s_Sfs (const Npp16s * pSrc, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit signed integer.

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.146.1.3 `NppStatus nppsNorm_Inf_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector C norm method

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.1.4 NppStatus nppsNorm\_Inf\_32fc32f (const Npp32fc \* pSrc, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)**

32-bit float complex vector C norm method, return value is 32-bit float.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormInfGetBufferSize\\_32fc32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.1.5 NppStatus nppsNorm\_Inf\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float vector C norm method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormInfGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.1.6 NppStatus nppsNorm\_Inf\_64fc64f (const Npp64fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float complex vector C norm method, return value is 64-bit float.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormInfGetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.1.7 NppStatus nppsNormInfGetBufferSize\_16s32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.146.1.8 NppStatus nppsNormInfGetBufferSize\_16s32s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.146.1.9 NppStatus nppsNormInfGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.146.1.10 NppStatus nppsNormInfGetBufferSize\_32fc32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_32fc32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.146.1.11 NppStatus nppsNormInfGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.146.1.12 NppStatus nppsNormInfGetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.147 L1 Norm

### Functions

- [NppStatus nppsNormL1GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_32f.*
- [NppStatus nppsNorm\\_L1\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector L1 norm method*
- [NppStatus nppsNormL1GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_64f.*
- [NppStatus nppsNorm\\_L1\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector L1 norm method*
- [NppStatus nppsNormL1GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s32f.*
- [NppStatus nppsNorm\\_L1\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L1 norm method, return value is 32-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_32fc64f.*
- [NppStatus nppsNorm\\_L1\\_32fc64f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector L1 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_64fc64f.*
- [NppStatus nppsNorm\\_L1\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector L1 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_L1\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.*
- [NppStatus nppsNormL1GetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s64s\_Sfs.*
- [NppStatus nppsNorm\\_L1\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

### 7.147.1 Function Documentation

#### 7.147.1.1 NppStatus nppsNorm\_L1\_16s32f (const Npp16s \* pSrc, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 32-bit float.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the L1 norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_16s32f](#) to determine the minimum number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.147.1.2 NppStatus nppsNorm\_L1\_16s32s\_Sfs (const Npp16s \* pSrc, int nLength, Npp32s \* pNorm, int nScaleFactor, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.147.1.3 NppStatus nppsNorm\_L1\_16s64s\_Sfs (const Npp16s \* pSrc, int nLength, Npp64s \* pNorm, int nScaleFactor, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL1GetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.1.4** `NppStatus nppsNorm_L1_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector L1 norm method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL1GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.1.5** `NppStatus nppsNorm_L1_32fc64f (const Npp32fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex vector L1 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL1GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.1.6** `NppStatus nppsNorm_L1_64f (const Npp64f * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float vector L1 norm method

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.147.1.7 NppStatus nppsNorm\_L1\_64fc64f (const Npp64fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

64-bit float complex vector L1 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.147.1.8 NppStatus nppsNormL1GetBufferSize\_16s32f (int nLength, int \* hpBufferSize)

Device-buffer size (in bytes) for nppsNorm\_L1\_16s32f.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

#### 7.147.1.9 NppStatus nppsNormL1GetBufferSize\_16s32s\_Sfs (int nLength, int \* hpBufferSize)

Device-buffer size (in bytes) for nppsNorm\_L1\_16s32s\_Sfs.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.147.1.10 NppStatus nppsNormL1GetBufferSize\_16s64s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L1\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.147.1.11 NppStatus nppsNormL1GetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L1\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.147.1.12 NppStatus nppsNormL1GetBufferSize\_32fc64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L1\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.147.1.13 NppStatus nppsNormL1GetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L1\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.147.1.14 NppStatus nppsNormL1GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.148 L2 Norm

### Functions

- [NppStatus nppsNormL2GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_32f.*
- [NppStatus nppsNorm\\_L2\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector L2 norm method*
- [NppStatus nppsNormL2GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_64f.*
- [NppStatus nppsNorm\\_L2\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector L2 norm method*
- [NppStatus nppsNormL2GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_16s32f.*
- [NppStatus nppsNorm\\_L2\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L2 norm method, return value is 32-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_32fc64f.*
- [NppStatus nppsNorm\\_L2\\_32fc64f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector L2 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_64fc64f.*
- [NppStatus nppsNorm\\_L2\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector L2 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_L2\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.*
- [NppStatus nppsNormL2SqrGetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2Sqr\_16s64s\_Sfs.*
- [NppStatus nppsNorm\\_L2Sqr\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

### 7.148.1 Function Documentation

#### 7.148.1.1 `NppStatus nppsNorm_L2_16s32f` (`const Npp16s * pSrc`, `int nLength`, `Npp32f * pNorm`, `Npp8u * pDeviceBuffer`)

16-bit signed short integer vector L2 norm method, return value is 32-bit float.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppsNormL2GetBufferSize_16s32f` to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.1.2 `NppStatus nppsNorm_L2_16s32s_Sfs` (`const Npp16s * pSrc`, `int nLength`, `Npp32s * pNorm`, `int nScaleFactor`, `Npp8u * pDeviceBuffer`)

16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppsNormL2GetBufferSize_16s32s_Sfs` to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.1.3 `NppStatus nppsNorm_L2_32f` (`const Npp32f * pSrc`, `int nLength`, `Npp32f * pNorm`, `Npp8u * pDeviceBuffer`)

32-bit float vector L2 norm method

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.1.4 NppStatus nppsNorm\_L2\_32fc64f (const Npp32fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

32-bit float complex vector L2 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.1.5 NppStatus nppsNorm\_L2\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float vector L2 norm method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.1.6 NppStatus nppsNorm\_L2\_64fc64f (const Npp64fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float complex vector L2 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.1.7** `NppStatus nppsNorm_L2Sqr_16s64s_Sfs (const Npp16s * pSrc, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2SqrGetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.1.8** `NppStatus nppsNormL2GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm\_L2\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.9** `NppStatus nppsNormL2GetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm\_L2\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.10 NppStatus nppsNormL2GetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.11 NppStatus nppsNormL2GetBufferSize\_32fc64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.12 NppStatus nppsNormL2GetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.13 NppStatus nppsNormL2GetBufferSize\_64fc64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.148.1.14 NppStatus nppsNormL2SqrGetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L2Sqr\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.149 Infinity Norm Diff

### Functions

- [NppStatus nppsNormDiffInfGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_32f.*
- [NppStatus nppsNormDiff\\_Inf\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float C norm method on two vectors' difference*
- [NppStatus nppsNormDiffInfGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_64f.*
- [NppStatus nppsNormDiff\\_Inf\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float C norm method on two vectors' difference*
- [NppStatus nppsNormDiffInfGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32f.*
- [NppStatus nppsNormDiff\\_Inf\\_16s32f](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_32fc32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_32fc32f.*
- [NppStatus nppsNormDiff\\_Inf\\_32fc32f](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_64fc64f.*
- [NppStatus nppsNormDiff\\_Inf\\_64fc64f](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32s\_Sfs.*
- [NppStatus nppsNormDiff\\_Inf\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.*

## 7.149.1 Function Documentation

### 7.149.1.1 `NppStatus nppsNormDiff_Inf_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.149.1.2 `NppStatus nppsNormDiff_Inf_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.149.1.3 `NppStatus nppsNormDiff_Inf_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float C norm method on two vectors' difference

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.1.4 NppStatus nppsNormDiff\_Inf\_32fc32f (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)**

32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_32fc32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.1.5 NppStatus nppsNormDiff\_Inf\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float C norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.149.1.6 `NppStatus nppsNormDiff_Inf_64fc64f (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppsNormDiffInfGetBufferSize_64fc64f` to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.149.1.7 `NppStatus nppsNormDiffInfGetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_Inf_16s32f`.

##### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

##### Returns:

NPP\_SUCCESS

#### 7.149.1.8 `NppStatus nppsNormDiffInfGetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_Inf_16s32s_Sfs`.

##### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

##### Returns:

NPP\_SUCCESS

#### 7.149.1.9 `NppStatus nppsNormDiffInfGetBufferSize_32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_Inf_32f`.

##### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.149.1.10 NppStatus nppsNormDiffInfGetBufferSize\_32fc32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsNormDiff\_Inf\_32fc32f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.149.1.11 NppStatus nppsNormDiffInfGetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsNormDiff\_Inf\_64f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.149.1.12 NppStatus nppsNormDiffInfGetBufferSize\_64fc64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsNormDiff\_Inf\_64fc64f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.150 L1 Norm Diff

### Functions

- [NppStatus nppsNormDiffL1GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_32f.*
- [NppStatus nppsNormDiff\\_L1\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float L1 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL1GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_64f.*
- [NppStatus nppsNormDiff\\_L1\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float L1 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL1GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32f.*
- [NppStatus nppsNormDiff\\_L1\\_16s32f](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffL1GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_32fc64f.*
- [NppStatus nppsNormDiff\\_L1\\_32fc64f](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL1GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_64fc64f.*
- [NppStatus nppsNormDiff\\_L1\\_64fc64f](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL1GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32s\_Sfs.*
- [NppStatus nppsNormDiff\\_L1\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.*
- [NppStatus nppsNormDiffL1GetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s64s\_Sfs.*
- [NppStatus nppsNormDiff\\_L1\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

### 7.150.1 Function Documentation

#### 7.150.1.1 `NppStatus nppsNormDiff_L1_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the L1 norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.150.1.2 `NppStatus nppsNormDiff_L1_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer..

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.150.1.3 `NppStatus nppsNormDiff_L1_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s64s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.150.1.4 NppStatus nppsNormDiff\_L1\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)

32-bit float L1 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.150.1.5 NppStatus nppsNormDiff\_L1\_32fc64f (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_32fc64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.1.6** `NppStatus nppsNormDiff_L1_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float L1 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.1.7** `NppStatus nppsNormDiff_L1_64fc64f (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.1.8** `NppStatus nppsNormDiffL1GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_L1_16s32f`.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.9 NppStatus nppsNormDiffL1GetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.10 NppStatus nppsNormDiffL1GetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.11 NppStatus nppsNormDiffL1GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.12 NppStatus nppsNormDiffL1GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.13 NppStatus nppsNormDiffL1GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.150.1.14 NppStatus nppsNormDiffL1GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.151 L2 Norm Diff

### Functions

- [NppStatus nppsNormDiffL2GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_32f.*
- [NppStatus nppsNormDiff\\_L2\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float L2 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL2GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_64f.*
- [NppStatus nppsNormDiff\\_L2\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float L2 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL2GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32f.*
- [NppStatus nppsNormDiff\\_L2\\_16s32f](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_32fc64f.*
- [NppStatus nppsNormDiff\\_L2\\_32fc64f](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_64fc64f.*
- [NppStatus nppsNormDiff\\_L2\\_64fc64f](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32s\_Sfs.*
- [NppStatus nppsNormDiff\\_L2\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.*
- [NppStatus nppsNormDiffL2SqrGetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2Sqr\_16s64s\_Sfs.*
- [NppStatus nppsNormDiff\\_L2Sqr\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

### 7.151.1 Function Documentation

#### 7.151.1.1 `NppStatus nppsNormDiff_L2_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.151.1.2 `NppStatus nppsNormDiff_L2_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.151.1.3 `NppStatus nppsNormDiff_L2_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float L2 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.151.1.4 `NppStatus nppsNormDiff_L2_32fc64f (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_32fc64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.151.1.5 `NppStatus nppsNormDiff_L2_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float L2 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.1.6** `NppStatus nppsNormDiff_L2_64fc64f (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppsNormDiffL2GetBufferSize_64fc64f` to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.1.7** `NppStatus nppsNormDiff_L2Sqr_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs` to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.1.8** `NppStatus nppsNormDiffL2GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_L2_16s32f`.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.9 NppStatus nppsNormDiffL2GetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.10 NppStatus nppsNormDiffL2GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.11 NppStatus nppsNormDiffL2GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.12 NppStatus nppsNormDiffL2GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.13 NppStatus nppsNormDiffL2GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.151.1.14 NppStatus nppsNormDiffL2SqrGetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2Sqr\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.152 Dot Product

### Functions

- [NppStatus nppsDotProdGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f.*
- [NppStatus nppsDotProd\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float dot product method, return value is 32-bit float.*
- [NppStatus nppsDotProdGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32fc.*
- [NppStatus nppsDotProd\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f32fc.*
- [NppStatus nppsDotProd\\_32f32fc](#) (const [Npp32f](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f64f.*
- [NppStatus nppsDotProd\\_32f64f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float dot product method, return value is 64-bit float.*
- [NppStatus nppsDotProdGetBufferSize\\_32fc64fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32fc64fc.*
- [NppStatus nppsDotProd\\_32fc64fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex dot product method, return value is 64-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f32fc64fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f32fc64fc.*
- [NppStatus nppsDotProd\\_32f32fc64fc](#) (const [Npp32f](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_64f.*
- [NppStatus nppsDotProd\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)

*64-bit float dot product method, return value is 64-bit float.*

- **NppStatus nppsDotProdGetBufferSize\_64fc** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_64fc.*
- **NppStatus nppsDotProd\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, int nLength, **Npp64fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
*64-bit float complex dot product method, return value is 64-bit float complex.*
- **NppStatus nppsDotProdGetBufferSize\_64f64fc** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_64f64fc.*
- **NppStatus nppsDotProd\_64f64fc** (const **Npp64f** \*pSrc1, const **Npp64fc** \*pSrc2, int nLength, **Npp64fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
*64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.*
- **NppStatus nppsDotProdGetBufferSize\_16s64s** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s64s.*
- **NppStatus nppsDotProd\_16s64s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp64s** \*pDp, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 64-bit signed integer.*
- **NppStatus nppsDotProdGetBufferSize\_16sc64sc** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc64sc.*
- **NppStatus nppsDotProd\_16sc64sc** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp64sc** \*pDp, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.*
- **NppStatus nppsDotProdGetBufferSize\_16s16sc64sc** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc64sc.*
- **NppStatus nppsDotProd\_16s16sc64sc** (const **Npp16s** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp64sc** \*pDp, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.*
- **NppStatus nppsDotProdGetBufferSize\_16s32f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s32f.*
- **NppStatus nppsDotProd\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp32f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 32-bit float.*
- **NppStatus nppsDotProdGetBufferSize\_16sc32fc** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc32fc.*
- **NppStatus nppsDotProd\_16sc32fc** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp32fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 32-bit float complex.*

- [NppStatus nppsDotProdGetBufferSize\\_16s16sc32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc32fc.*
- [NppStatus nppsDotProd\\_16s16sc32fc](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s\_Sfs.*
- [NppStatus nppsDotProd\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp16s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 16-bit signed short integer.*
- [NppStatus nppsDotProdGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc\_Sfs.*
- [NppStatus nppsDotProd\\_16sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp16sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32s\_Sfs.*
- [NppStatus nppsDotProd\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc1, const [Npp32s](#) \*pSrc2, int nLength, [Npp32s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed integer dot product method, return value is 32-bit signed integer.*
- [NppStatus nppsDotProdGetBufferSize\\_32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32sc\_Sfs.*
- [NppStatus nppsDotProd\\_32sc\\_Sfs](#) (const [Npp32sc](#) \*pSrc1, const [Npp32sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s32s\_Sfs.*
- [NppStatus nppsDotProd\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 32-bit signed integer.*
- [NppStatus nppsDotProdGetBufferSize\\_16s16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc32sc\_Sfs.*
- [NppStatus nppsDotProd\\_16s16sc32sc\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*

- **NppStatus nppsDotProdGetBufferSize\_16s32s32s\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s32s32s\_Sfs.*
- **NppStatus nppsDotProd\_16s32s32s\_Sfs** (const Npp16s \*pSrc1, const Npp32s \*pSrc2, int nLength, Npp32s \*pDp, int nScaleFactor, Npp8u \*pDeviceBuffer)  
*16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.*
- **NppStatus nppsDotProdGetBufferSize\_16s16sc\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc\_Sfs.*
- **NppStatus nppsDotProd\_16s16sc\_Sfs** (const Npp16s \*pSrc1, const Npp16sc \*pSrc2, int nLength, Npp16sc \*pDp, int nScaleFactor, Npp8u \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.*
- **NppStatus nppsDotProdGetBufferSize\_16sc32sc\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc32sc\_Sfs.*
- **NppStatus nppsDotProd\_16sc32sc\_Sfs** (const Npp16sc \*pSrc1, const Npp16sc \*pSrc2, int nLength, Npp32sc \*pDp, int nScaleFactor, Npp8u \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*
- **NppStatus nppsDotProdGetBufferSize\_32s32sc\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32s32sc\_Sfs.*
- **NppStatus nppsDotProd\_32s32sc\_Sfs** (const Npp32s \*pSrc1, const Npp32sc \*pSrc2, int nLength, Npp32sc \*pDp, int nScaleFactor, Npp8u \*pDeviceBuffer)  
*32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*

## 7.152.1 Function Documentation

### 7.152.1.1 NppStatus nppsDotProd\_16s16sc32fc (const Npp16s \*pSrc1, const Npp16sc \*pSrc2, int nLength, Npp32fc \*pDp, Npp8u \*pDeviceBuffer)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.

#### Parameters:

**pSrc1** Source Signal Pointer.

**pSrc2** Source Signal Pointer.

**nLength** Signal Length.

**pDp** Pointer to the dot product result.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc32fc](#) to determine the minimum number of bytes required.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.1.2 NppStatus nppsDotProd\_16s16sc32sc\_Sfs (const Npp16s \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp32sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppsDotProdGetBufferSize\\_16s16sc32sc\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.1.3 NppStatus nppsDotProd\_16s16sc64sc (const Npp16s \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64sc \* pDp, Npp8u \* pDeviceBuffer)**

16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppsDotProdGetBufferSize\\_16s16sc64sc](#) to determine the minium number of bytes required.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.1.4 NppStatus nppsDotProd\_16s16sc\_Sfs (const Npp16s \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp16sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.5 NppStatus nppsDotProd\_16s32f (const Npp16s \* pSrc1, const Npp16s \* pSrc2, int nLength, Npp32f \* pDp, Npp8u \* pDeviceBuffer)**

16-bit signed short integer dot product method, return value is 32-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.6 NppStatus nppsDotProd\_16s32s32s\_Sfs (const Npp16s \* pSrc1, const Npp32s \* pSrc2, int nLength, Npp32s \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32s32s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.7** `NppStatus nppsDotProd_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pDp, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 32-bit signed integer.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.8** `NppStatus nppsDotProd_16s64s (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pDp, Npp8u * pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 64-bit signed integer.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s64s](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.9 NppStatus nppsDotProd\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, int nLength, Npp16s \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer dot product method, return value is 16-bit signed short integer.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.10 NppStatus nppsDotProd\_16sc32fc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp32fc \* pDp, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 32-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.11 NppStatus nppsDotProd\_16sc32sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp32sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc32sc\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.152.1.12 **NppStatus nppsDotProd\_16sc64sc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64sc \* pDp, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc64sc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.152.1.13 **NppStatus nppsDotProd\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp16sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.14 NppStatus nppsDotProd\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, int nLength, Npp32f \* pDp, Npp8u \* pDeviceBuffer)**

32-bit float dot product method, return value is 32-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.15 NppStatus nppsDotProd\_32f32fc (const Npp32f \* pSrc1, const Npp32fc \* pSrc2, int nLength, Npp32fc \* pDp, Npp8u \* pDeviceBuffer)**

32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f32fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.16 NppStatus nppsDotProd\_32f32fc64fc (const Npp32f \* pSrc1, const Npp32fc \* pSrc2, int nLength, Npp64fc \* pDp, Npp8u \* pDeviceBuffer)**

32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f32fc64fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.17** `NppStatus nppsDotProd_32f64f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp64f * pDp, Npp8u * pDeviceBuffer)`

32-bit float dot product method, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.18** `NppStatus nppsDotProd_32fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp32fc * pDp, Npp8u * pDeviceBuffer)`

32-bit float complex dot product method, return value is 32-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.19** `NppStatus nppsDotProd_32fc64fc (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64fc * pDp, Npp8u * pDeviceBuffer)`

32-bit float complex dot product method, return value is 64-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32fc64fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.20 NppStatus nppsDotProd\_32s32sc\_Sfs (const Npp32s \* pSrc1, const Npp32sc \* pSrc2, int nLength, Npp32sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32s32sc\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.21 NppStatus nppsDotProd\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, int nLength, Npp32s \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

32-bit signed integer dot product method, return value is 32-bit signed integer.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.22 NppStatus nppsDotProd\_32sc\_Sfs (const Npp32sc \* pSrc1, const Npp32sc \* pSrc2, int nLength, Npp32sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32sc\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.23 NppStatus nppsDotProd\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, int nLength, Npp64f \* pDp, Npp8u \* pDeviceBuffer)**

64-bit float dot product method, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64f](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.24 NppStatus nppsDotProd\_64f64fc (const Npp64f \* pSrc1, const Npp64fc \* pSrc2, int nLength, Npp64fc \* pDp, Npp8u \* pDeviceBuffer)**

64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64f64fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.25 NppStatus nppsDotProd\_64fc (const Npp64fc \* pSrc1, const Npp64fc \* pSrc2, int nLength, Npp64fc \* pDp, Npp8u \* pDeviceBuffer)**

64-bit float complex dot product method, return value is 64-bit float complex.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64fc](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.152.1.26 NppStatus nppsDotProdGetBufferSize\_16s16sc32fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s16sc32fc.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.27 NppStatus nppsDotProdGetBufferSize\_16s16sc32sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s16sc32sc\_Sfs.

**Parameters:**

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.28 NppStatus nppsDotProdGetBufferSize\_16s16sc64sc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s16sc64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.29 NppStatus nppsDotProdGetBufferSize\_16s16sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsDotProd\_16s16sc\_Sfs*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.30 NppStatus nppsDotProdGetBufferSize\_16s32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsDotProd\_16s32f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.31 NppStatus nppsDotProdGetBufferSize\_16s32s32s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for *nppsDotProd\_16s32s32s\_Sfs*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.32 NppStatus nppsDotProdGetBufferSize\_16s32s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.33 NppStatus nppsDotProdGetBufferSize\_16s64s (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.34 NppStatus nppsDotProdGetBufferSize\_16s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.35 NppStatus nppsDotProdGetBufferSize\_16sc32fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16sc32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.36 NppStatus nppsDotProdGetBufferSize\_16sc32sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16sc32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.37 NppStatus nppsDotProdGetBufferSize\_16sc64sc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16sc64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.38 NppStatus nppsDotProdGetBufferSize\_16sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.39 NppStatus nppsDotProdGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.40 NppStatus nppsDotProdGetBufferSize\_32f32fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32f32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.41 NppStatus nppsDotProdGetBufferSize\_32f32fc64fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32f32fc64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.42 NppStatus nppsDotProdGetBufferSize\_32f64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32f64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.43 NppStatus nppsDotProdGetBufferSize\_32fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.44 NppStatus nppsDotProdGetBufferSize\_32fc64fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32fc64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.45 NppStatus nppsDotProdGetBufferSize\_32s32sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32s32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.46 NppStatus nppsDotProdGetBufferSize\_32s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.47 NppStatus nppsDotProdGetBufferSize\_32sc\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.48 NppStatus nppsDotProdGetBufferSize\_64f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.49 NppStatus nppsDotProdGetBufferSize\_64f64fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_64f64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.152.1.50 NppStatus nppsDotProdGetBufferSize\_64fc (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsDotProd\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.153 Count In Range

### Functions

- [NppStatus nppsCountInRangeGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsCountInRange\_32s.*
- [NppStatus nppsCountInRange\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, int \*pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) \*pDeviceBuffer)  
*Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.*

### 7.153.1 Function Documentation

#### 7.153.1.1 [NppStatus nppsCountInRange\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, int \*pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) \*pDeviceBuffer)

Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pCounts* Pointer to the number of elements.

*nLowerBound* Lower bound of the specified range.

*nUpperBound* Upper bound of the specified range.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsCountInRangeGetBufferSize\\_32s](#) to determine the minimum number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.153.1.2 [NppStatus nppsCountInRangeGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)

Device-buffer size (in bytes) for nppsCountInRange\_32s.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

#### Returns:

NPP\_SUCCESS

## 7.154 Count Zero Crossings

### Functions

- [NppStatus nppsZeroCrossingGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsZeroCrossing\_16s32f.*
- [NppStatus nppsZeroCrossing\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer zero crossing method, return value is 32-bit floating point.*
- [NppStatus nppsZeroCrossingGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsZeroCrossing\_32f.*
- [NppStatus nppsZeroCrossing\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating-point zero crossing method, return value is 32-bit floating point.*

### 7.154.1 Function Documentation

#### 7.154.1.1 [NppStatus nppsZeroCrossing\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer zero crossing method, return value is 32-bit floating point.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pValZC* [Pointer to the output result](#).

*tZCType* [Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppsZeroCrossingGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.154.1.2 [NppStatus nppsZeroCrossing\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)

32-bit floating-point zero crossing method, return value is 32-bit floating point.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pValZC* [Pointer to the output result](#).

*tZCType* Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsZeroCrossingGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.154.1.3 NppStatus nppsZeroCrossingGetBufferSize\_16s32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsZeroCrossing\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.154.1.4 NppStatus nppsZeroCrossingGetBufferSize\_32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsZeroCrossing\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## **7.155 Memory Management**



# Chapter 8

## Data Structure Documentation

### 8.1 Npp16sc Struct Reference

Complex Number This struct represents a short complex number.

```
#include <nppdefs.h>
```

#### Data Fields

- [Npp16s re](#)  
*Real part.*
- [Npp16s im](#)  
*Imaginary part.*

#### 8.1.1 Detailed Description

Complex Number This struct represents a short complex number.

#### 8.1.2 Field Documentation

##### 8.1.2.1 Npp16s Npp16sc::im

Imaginary part.

##### 8.1.2.2 Npp16s Npp16sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.2 Npp16uc Struct Reference

Complex Number This struct represents an unsigned short complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp16u re](#)  
*Real part.*
- [Npp16u im](#)  
*Imaginary part.*

### 8.2.1 Detailed Description

Complex Number This struct represents an unsigned short complex number.

### 8.2.2 Field Documentation

#### 8.2.2.1 Npp16u Npp16uc::im

Imaginary part.

#### 8.2.2.2 Npp16u Npp16uc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.3 Npp32fc Struct Reference

Complex Number This struct represents a single floating-point complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32f re](#)  
*Real part.*
- [Npp32f im](#)  
*Imaginary part.*

### 8.3.1 Detailed Description

Complex Number This struct represents a single floating-point complex number.

### 8.3.2 Field Documentation

#### 8.3.2.1 Npp32f Npp32fc::im

Imaginary part.

#### 8.3.2.2 Npp32f Npp32fc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.4 Npp32sc Struct Reference

Complex Number This struct represents a signed int complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32s re](#)  
*Real part.*
- [Npp32s im](#)  
*Imaginary part.*

### 8.4.1 Detailed Description

Complex Number This struct represents a signed int complex number.

### 8.4.2 Field Documentation

#### 8.4.2.1 Npp32s Npp32sc::im

Imaginary part.

#### 8.4.2.2 Npp32s Npp32sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.5 Npp32uc Struct Reference

Complex Number This struct represents an unsigned int complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32u re](#)  
*Real part.*
- [Npp32u im](#)  
*Imaginary part.*

### 8.5.1 Detailed Description

Complex Number This struct represents an unsigned int complex number.

### 8.5.2 Field Documentation

#### 8.5.2.1 Npp32u Npp32uc::im

Imaginary part.

#### 8.5.2.2 Npp32u Npp32uc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.6 Npp64fc Struct Reference

Complex Number This struct represents a double floating-point complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp64f re](#)  
*Real part.*
- [Npp64f im](#)  
*Imaginary part.*

### 8.6.1 Detailed Description

Complex Number This struct represents a double floating-point complex number.

### 8.6.2 Field Documentation

#### 8.6.2.1 Npp64f Npp64fc::im

Imaginary part.

#### 8.6.2.2 Npp64f Npp64fc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.7 Npp64sc Struct Reference

Complex Number This struct represents a long long complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp64s re](#)  
*Real part.*
- [Npp64s im](#)  
*Imaginary part.*

### 8.7.1 Detailed Description

Complex Number This struct represents a long long complex number.

### 8.7.2 Field Documentation

#### 8.7.2.1 Npp64s Npp64sc::im

Imaginary part.

#### 8.7.2.2 Npp64s Npp64sc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.8 Npp8uc Struct Reference

Complex Number This struct represents an unsigned char complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp8u re](#)  
*Real part.*
- [Npp8u im](#)  
*Imaginary part.*

### 8.8.1 Detailed Description

Complex Number This struct represents an unsigned char complex number.

### 8.8.2 Field Documentation

#### 8.8.2.1 Npp8u Npp8uc::im

Imaginary part.

#### 8.8.2.2 Npp8u Npp8uc::re

Real part.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h

## 8.9 NppiHaarBuffer Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int `haarBufferSize`  
*size of the buffer*
- `Npp32s * haarBuffer`  
*buffer*

### 8.9.1 Field Documentation

#### 8.9.1.1 `Npp32s* NppiHaarBuffer::haarBuffer`

`buffer`

#### 8.9.1.2 `int NppiHaarBuffer::haarBufferSize`

*size of the buffer*

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h`

## 8.10 NppiHaarClassifier\_32f Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int `numClassifiers`  
*number of classifiers*
- `Npp32s * classifiers`  
*packed classifier data 40 bytes each*
- `size_t classifierStep`
- `NppiSize classifierSize`
- `Npp32s * counterDevice`

### 8.10.1 Field Documentation

#### 8.10.1.1 `Npp32s* NppiHaarClassifier_32f::classifiers`

packed classifier data 40 bytes each

#### 8.10.1.2 `NppiSize NppiHaarClassifier_32f::classifierSize`

#### 8.10.1.3 `size_t NppiHaarClassifier_32f::classifierStep`

#### 8.10.1.4 `Npp32s* NppiHaarClassifier_32f::counterDevice`

#### 8.10.1.5 `int NppiHaarClassifier_32f::numClassifiers`

number of classifiers

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h`

## 8.11 NppiPoint Struct Reference

2D Point

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate.*
- `int y`  
*y-coordinate.*

### 8.11.1 Detailed Description

2D Point

### 8.11.2 Field Documentation

#### 8.11.2.1 `int NppiPoint::x`

x-coordinate.

#### 8.11.2.2 `int NppiPoint::y`

y-coordinate.

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h`

## 8.12 NppiRect Struct Reference

2D Rectangle This struct contains position and size information of a rectangle in two space.

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate of upper left corner.*
- `int y`  
*y-coordinate of upper left corner.*
- `int width`  
*Rectangle width.*
- `int height`  
*Rectangle height.*

### 8.12.1 Detailed Description

2D Rectangle This struct contains position and size information of a rectangle in two space.

The rectangle's position is usually signified by the coordinate of its upper-left corner.

### 8.12.2 Field Documentation

#### 8.12.2.1 `int NppiRect::height`

Rectangle height.

#### 8.12.2.2 `int NppiRect::width`

Rectangle width.

#### 8.12.2.3 `int NppiRect::x`

x-coordinate of upper left corner.

#### 8.12.2.4 `int NppiRect::y`

y-coordinate of upper left corner.

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h`

## 8.13 NppiSize Struct Reference

2D Size This struct typically represents the size of a rectangular region in two space.

```
#include <nppdefs.h>
```

### Data Fields

- `int width`  
*Rectangle width.*
- `int height`  
*Rectangle height.*

### 8.13.1 Detailed Description

2D Size This struct typically represents the size of a rectangular region in two space.

### 8.13.2 Field Documentation

#### 8.13.2.1 `int NppiSize::height`

Rectangle height.

#### 8.13.2.2 `int NppiSize::width`

Rectangle width.

The documentation for this struct was generated from the following file:

- `C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h`

## 8.14 NppLibraryVersion Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [major](#)  
*Major version number.*
- int [minor](#)  
*Minor version number.*
- int [build](#)  
*Build number. This reflects the nightly build this release was made from.*

### 8.14.1 Field Documentation

#### 8.14.1.1 int NppLibraryVersion::build

Build number. This reflects the nightly build this release was made from.

#### 8.14.1.2 int NppLibraryVersion::major

Major version number.

#### 8.14.1.3 int NppLibraryVersion::minor

Minor version number.

The documentation for this struct was generated from the following file:

- C:/Perforce/sw/rel/gpgpu/toolkit/r5.0/NPP/npp/include/nppdefs.h