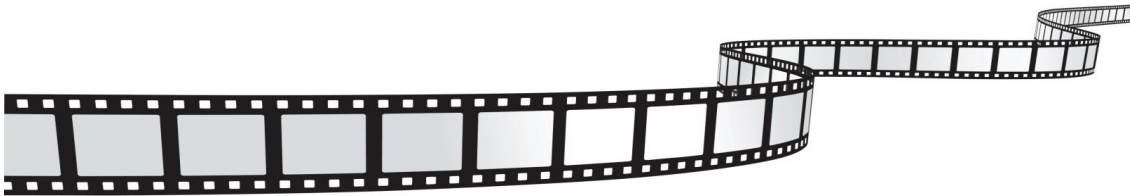


18.08.10



# Contents

0.1	Introduction . . . . .	9
0.2	Data Structure Index . . . . .	9
0.2.1	Data Structures . . . . .	9
0.3	File Index . . . . .	10
0.3.1	File List . . . . .	10
0.4	Data Structure Documentation . . . . .	15
0.4.1	bitbuffer_s Struct Reference . . . . .	15
0.4.2	fundamental_debug Struct Reference . . . . .	16
0.4.3	huffman_s Struct Reference . . . . .	16
0.4.4	mvcvPatternPoint Struct Reference . . . . .	17
0.4.5	RANSAC_Inputs Struct Reference . . . . .	18
0.4.6	RANSAC_Outputs Struct Reference . . . . .	18
0.4.7	ransac_params Struct Reference . . . . .	19
0.5	File Documentation . . . . .	19
0.5.1	absoluteDiff.h File Reference . . . . .	19
0.5.2	accumulateFp16.h File Reference . . . . .	19
0.5.3	accumulateFp16MaxPool2s1.h File Reference . . . . .	19
0.5.4	accumulateFp16MaxPool2s2.h File Reference . . . . .	20
0.5.5	accumulateFp16MaxPool3s1.h File Reference . . . . .	20
0.5.6	accumulateFp16MaxPool3s2.h File Reference . . . . .	20
0.5.7	accumulateFp16MaxPool3s2withReLU.h File Reference . . . . .	20
0.5.8	accumulateSquare.h File Reference . . . . .	20
0.5.9	accumulateWeighted.h File Reference . . . . .	20
0.5.10	addV2Fp16.h File Reference . . . . .	20
0.5.11	aggregateAllPaths64.h File Reference . . . . .	20
0.5.12	aggregateCostSGBM32_clamp.h File Reference . . . . .	21
0.5.13	aggregateCostSGBM64.h File Reference . . . . .	21
0.5.14	aggregateCostSGBM64_clamp.h File Reference . . . . .	21
0.5.15	aggregateFivePaths32.h File Reference . . . . .	21

0.5.16	<a href="#">aggregateFivePaths64.h File Reference</a>	21
0.5.17	<a href="#">aggregateThreePaths32.h File Reference</a>	21
0.5.18	<a href="#">aggregateThreePaths64.h File Reference</a>	21
0.5.19	<a href="#">app_config.h File Reference</a>	22
0.5.20	<a href="#">arithmeticAdd.h File Reference</a>	23
0.5.21	<a href="#">arithmeticAddmask.h File Reference</a>	23
0.5.22	<a href="#">arithmeticSub.h File Reference</a>	23
0.5.23	<a href="#">arithmeticSubFp16ToFp16.h File Reference</a>	23
0.5.24	<a href="#">arithmeticSubmask.h File Reference</a>	23
0.5.25	<a href="#">arithmeticSubU16.h File Reference</a>	23
0.5.26	<a href="#">averageV3.h File Reference</a>	23
0.5.27	<a href="#">avgPool7x7xk.h File Reference</a>	24
0.5.28	<a href="#">bilateral5x5.h File Reference</a>	24
0.5.29	<a href="#">bilinearInterpolation.h File Reference</a>	24
0.5.30	<a href="#">bilinearInterpolation_u16.h File Reference</a>	24
0.5.31	<a href="#">bilinearInterpolationAligned_u16.h File Reference</a>	24
0.5.32	<a href="#">bitwiseAnd.h File Reference</a>	24
0.5.33	<a href="#">bitwiseAndMask.h File Reference</a>	24
0.5.34	<a href="#">bitwiseNot.h File Reference</a>	25
0.5.35	<a href="#">bitwiseOr.h File Reference</a>	25
0.5.36	<a href="#">bitwiseOrMask.h File Reference</a>	25
0.5.37	<a href="#">bitwiseXor.h File Reference</a>	25
0.5.38	<a href="#">bitwiseXorMask.h File Reference</a>	25
0.5.39	<a href="#">boxFilter.h File Reference</a>	25
0.5.40	<a href="#">boxFilter11x11.h File Reference</a>	25
0.5.41	<a href="#">boxFilter13x13.h File Reference</a>	25
0.5.42	<a href="#">boxFilter15x15.h File Reference</a>	26
0.5.43	<a href="#">boxFilter3x3.h File Reference</a>	26
0.5.44	<a href="#">boxFilter5x5.h File Reference</a>	26
0.5.45	<a href="#">boxFilter7x7.h File Reference</a>	26
0.5.46	<a href="#">boxFilter9x9.h File Reference</a>	26
0.5.47	<a href="#">Buffers.h File Reference</a>	26
0.5.48	<a href="#">calcBxBy.h File Reference</a>	28
0.5.49	<a href="#">calcEpipolarDistance.h File Reference</a>	28
0.5.50	<a href="#">calcG.h File Reference</a>	28
0.5.51	<a href="#">canny.h File Reference</a>	28

0.5.52	<a href="#">censusMatching16.h File Reference</a>	28
0.5.53	<a href="#">censusMatching32.h File Reference</a>	28
0.5.54	<a href="#">censusMatching64.h File Reference</a>	28
0.5.55	<a href="#">censusMatching65.h File Reference</a>	28
0.5.56	<a href="#">censusMatchingPyr.h File Reference</a>	29
0.5.57	<a href="#">censusMatchingPyrOnePos.h File Reference</a>	29
0.5.58	<a href="#">censusMatchingPyrOnePosWindow.h File Reference</a>	29
0.5.59	<a href="#">censusMin16.h File Reference</a>	29
0.5.60	<a href="#">censusMin3.h File Reference</a>	29
0.5.61	<a href="#">censusMin32.h File Reference</a>	29
0.5.62	<a href="#">censusMin64.h File Reference</a>	29
0.5.63	<a href="#">censusMin65.h File Reference</a>	30
0.5.64	<a href="#">censusMin7.h File Reference</a>	30
0.5.65	<a href="#">censusMinConfidence32.h File Reference</a>	30
0.5.66	<a href="#">censusMinConfidence64.h File Reference</a>	31
0.5.67	<a href="#">censusMinConfidence64Subpixel.h File Reference</a>	31
0.5.68	<a href="#">censusMinSubpixel3.h File Reference</a>	32
0.5.69	<a href="#">censusMinSubpixel3Window.h File Reference</a>	32
0.5.70	<a href="#">censusTransform11x11.h File Reference</a>	32
0.5.71	<a href="#">censusTransform11x11u8.h File Reference</a>	32
0.5.72	<a href="#">censusTransform5x5.h File Reference</a>	32
0.5.73	<a href="#">censusTransformAverageRef7x7.h File Reference</a>	32
0.5.74	<a href="#">censusTransformAverageRefMask7x7.h File Reference</a>	32
0.5.75	<a href="#">channelExtract.h File Reference</a>	32
0.5.76	<a href="#">computeAD32.h File Reference</a>	33
0.5.77	<a href="#">computeAD64.h File Reference</a>	33
0.5.78	<a href="#">computeADPyrOnePos.h File Reference</a>	33
0.5.79	<a href="#">computeADPyrOnePosWindow.h File Reference</a>	33
0.5.80	<a href="#">computeCombinedCost3.h File Reference</a>	33
0.5.81	<a href="#">computeCombinedCost32.h File Reference</a>	33
0.5.82	<a href="#">computeCombinedCost3Window.h File Reference</a>	33
0.5.83	<a href="#">computeCombinedCost64.h File Reference</a>	33
0.5.84	<a href="#">conv3x3fp32Scharr.h File Reference</a>	34
0.5.85	<a href="#">convert_fp16_u8.h File Reference</a>	34
0.5.86	<a href="#">convert_u8_fp16.h File Reference</a>	34
0.5.87	<a href="#">convertFp16ToQ4.h File Reference</a>	34

0.5.88	<a href="#">convertFrom12BppTo8Bpp.h File Reference</a>	34
0.5.89	<a href="#">convertYUV400ToYUV422.h File Reference</a>	34
0.5.90	<a href="#">convolution1x11.h File Reference</a>	34
0.5.91	<a href="#">convolution1x11s1hhhh.h File Reference</a>	35
0.5.92	<a href="#">convolution1x11s1xxhx.h File Reference</a>	35
0.5.93	<a href="#">convolution1x11s2hhhh.h File Reference</a>	35
0.5.94	<a href="#">convolution1x11s2xxhx.h File Reference</a>	35
0.5.95	<a href="#">convolution1x11s3hhhh.h File Reference</a>	35
0.5.96	<a href="#">convolution1x11s4hhhh.h File Reference</a>	35
0.5.97	<a href="#">convolution1x11s8hhhh.h File Reference</a>	36
0.5.98	<a href="#">convolution15x1.h File Reference</a>	36
0.5.99	<a href="#">convolution1x15.h File Reference</a>	36
0.5.100	<a href="#">convolution1x5.h File Reference</a>	36
0.5.101	<a href="#">convolution1x5Fp16ToFp16.h File Reference</a>	36
0.5.102	<a href="#">convolution1x7.h File Reference</a>	36
0.5.103	<a href="#">convolution1x7Fp16ToFp16.h File Reference</a>	36
0.5.104	<a href="#">convolution1x9.h File Reference</a>	36
0.5.105	<a href="#">convolution3x3.h File Reference</a>	37
0.5.106	<a href="#">convolution3x3Fp16ToFp16.h File Reference</a>	37
0.5.107	<a href="#">convolution3x3s2hhhh.h File Reference</a>	37
0.5.108	<a href="#">convolution3x3s2xxhx.h File Reference</a>	37
0.5.109	<a href="#">convolution3x3s3hhhh.h File Reference</a>	37
0.5.110	<a href="#">convolution3x3s4hhhh.h File Reference</a>	37
0.5.111	<a href="#">convolution3x3s8hhhh.h File Reference</a>	37
0.5.112	<a href="#">convolution5x1.h File Reference</a>	38
0.5.113	<a href="#">convolution5x1Fp16ToFp16.h File Reference</a>	38
0.5.114	<a href="#">convolution5x5.h File Reference</a>	38
0.5.115	<a href="#">convolution5x5Fp16ToFp16.h File Reference</a>	38
0.5.116	<a href="#">convolution5x5s2hhhh.h File Reference</a>	38
0.5.117	<a href="#">convolution5x5s3hhhh.h File Reference</a>	38
0.5.118	<a href="#">convolution5x5s4hhhh.h File Reference</a>	38
0.5.119	<a href="#">convolution5x5s8hhhh.h File Reference</a>	38
0.5.120	<a href="#">convolution7x1.h File Reference</a>	39
0.5.121	<a href="#">convolution7x1Fp16ToFp16.h File Reference</a>	39
0.5.122	<a href="#">convolution7x7.h File Reference</a>	39
0.5.123	<a href="#">convolution7x7Fp16ToFp16.h File Reference</a>	39

0.5.124 convolution7x7Fp16ToU8.h File Reference . . . . .	39
0.5.125 convolution7x7s2hhhh.h File Reference . . . . .	39
0.5.126 convolution7x7s3hhhh.h File Reference . . . . .	39
0.5.127 convolution7x7s4hhhh.h File Reference . . . . .	39
0.5.128 convolution7x7s8hhhh.h File Reference . . . . .	40
0.5.129 convolution9x1.h File Reference . . . . .	40
0.5.130 convolution9x9.h File Reference . . . . .	40
0.5.131 convolution9x9Fp16ToFp16.h File Reference . . . . .	40
0.5.132 convolution9x9s2hhhh.h File Reference . . . . .	40
0.5.133 convolution9x9s3hhhh.h File Reference . . . . .	40
0.5.134 convolution9x9s4hhhh.h File Reference . . . . .	40
0.5.135 convolution9x9s8hhhh.h File Reference . . . . .	40
0.5.136 convSeparable11x11.h File Reference . . . . .	41
0.5.137 convSeparable11x11Fp16ToFp16.h File Reference . . . . .	41
0.5.138 convSeparable3x3.h File Reference . . . . .	41
0.5.139 convSeparable3x3Fp16ToFp16.h File Reference . . . . .	41
0.5.140 convSeparable5x5.h File Reference . . . . .	41
0.5.141 convSeparable5x5Fp16ToFp16.h File Reference . . . . .	41
0.5.142 convSeparable7x7.h File Reference . . . . .	41
0.5.143 convSeparable7x7Fp16ToFp16.h File Reference . . . . .	41
0.5.144 convSeparable9x9.h File Reference . . . . .	42
0.5.145 convSeparable9x9Fp16ToFp16.h File Reference . . . . .	42
0.5.146 cornerMinEigenVal.h File Reference . . . . .	42
0.5.147 cornerMinEigenVal_fp32.h File Reference . . . . .	42
0.5.148 cornerMinEigenVal_patched.h File Reference . . . . .	42
0.5.149 cvtColorKernelChromaYUV420ToNV12.h File Reference . . . . .	42
0.5.150 cvtColorKernelChromaYUV444ToNV12.h File Reference . . . . .	42
0.5.151 cvtColorKernelRGBToYUV.h File Reference . . . . .	43
0.5.152 cvtColorKernelRGBToYUV422.h File Reference . . . . .	43
0.5.153 cvtColorKernelYUV422ToRGB.h File Reference . . . . .	43
0.5.154 cvtColorKernelYUVToRGB.h File Reference . . . . .	43
0.5.155 cvtColorNV21toRGB.h File Reference . . . . .	43
0.5.156 cvtColorRGBfp16ToLumaU8.h File Reference . . . . .	43
0.5.157 cvtColorRGBfp16ToUV420U8.h File Reference . . . . .	43
0.5.158 cvtColorRGBtoChromaNV12.h File Reference . . . . .	44
0.5.159 cvtColorRGBtoLuma.h File Reference . . . . .	44

0.5.160 cvtColorRGBtoLumaNV12.h File Reference . . . . .	44
0.5.161 cvtColorRGBtoNV12.h File Reference . . . . .	44
0.5.162 cvtColorRGBtoNV21.h File Reference . . . . .	44
0.5.163 cvtColorRGBtoUV.h File Reference . . . . .	44
0.5.164 cvtColorRGBtoUV420.h File Reference . . . . .	44
0.5.165 data.h File Reference . . . . .	44
0.5.166 dct16bit.h File Reference . . . . .	44
0.5.167 Defines.h File Reference . . . . .	45
0.5.168 dilate.h File Reference . . . . .	45
0.5.169 dilate3x3.h File Reference . . . . .	45
0.5.170 dilate5x5.h File Reference . . . . .	45
0.5.171 dilate7x7.h File Reference . . . . .	46
0.5.172 disp2depth.h File Reference . . . . .	46
0.5.173 downsampleBilinearLine.h File Reference . . . . .	46
0.5.174 eigenValVec4x4Array.h File Reference . . . . .	46
0.5.175 equalizeHist.h File Reference . . . . .	46
0.5.176 erode3x3.h File Reference . . . . .	46
0.5.177 erode5x5.h File Reference . . . . .	46
0.5.178 erode7x7.h File Reference . . . . .	46
0.5.179 f32UnitTestSample.h File Reference . . . . .	47
0.5.180 fast9M2.h File Reference . . . . .	47
0.5.181 fast9ScoreCv.h File Reference . . . . .	47
0.5.182 fast9u16score.h File Reference . . . . .	47
0.5.183 fastAtan2Positive.h File Reference . . . . .	47
0.5.184 flipKernel.h File Reference . . . . .	48
0.5.185 fm8Point.h File Reference . . . . .	48
0.5.186 FundamentalMatrix.h File Reference . . . . .	48
0.5.187 gauss.h File Reference . . . . .	49
0.5.188 gauss1x5_u16in_u32out.h File Reference . . . . .	49
0.5.189 gauss5x1_u32in_u16out.h File Reference . . . . .	49
0.5.190 gaussHx2.h File Reference . . . . .	49
0.5.191 gaussHx2_fp16.h File Reference . . . . .	49
0.5.192 gaussVx2.h File Reference . . . . .	49
0.5.193 gaussVx2_fp16.h File Reference . . . . .	50
0.5.194 hammingDistance.h File Reference . . . . .	50
0.5.195 harrisResponse.h File Reference . . . . .	50

0.5.196 histogram.h File Reference . . . . .	50
0.5.197 huffman_encode.h File Reference . . . . .	50
0.5.198 integralImageSquareSumFloatM2.h File Reference . . . . .	51
0.5.199 integralImageSquareSumM2.h File Reference . . . . .	51
0.5.200 integralImageSumFloatM2.h File Reference . . . . .	51
0.5.201 integralImageSumM2.h File Reference . . . . .	51
0.5.202 integralImageSumU16U32.h File Reference . . . . .	51
0.5.203 interpolatePixelBilinear.h File Reference . . . . .	51
0.5.204 interpolatePixelBilinearS16.h File Reference . . . . .	52
0.5.205 jpegGetBlockY420.h File Reference . . . . .	52
0.5.206 kernels.h File Reference . . . . .	52
0.5.207 kernelUnitTestSample.h File Reference . . . . .	53
0.5.208 localMaxMin3x3_s16.h File Reference . . . . .	54
0.5.209 lookupTable10to16.h File Reference . . . . .	55
0.5.210 lookupTable10to8.h File Reference . . . . .	55
0.5.211 lookupTable12to16.h File Reference . . . . .	55
0.5.212 lookupTable12to8.h File Reference . . . . .	55
0.5.213 lookupTable8to10.h File Reference . . . . .	55
0.5.214 lookupTable8to8.h File Reference . . . . .	55
0.5.215 matmulBT_xxii.h File Reference . . . . .	56
0.5.216 matrixInv3x3_fp32.h File Reference . . . . .	56
0.5.217 matrixInv4x4_fp32.h File Reference . . . . .	56
0.5.218 matrixVectorMultfp16x4.h File Reference . . . . .	56
0.5.219 maximumV2.h File Reference . . . . .	56
0.5.220 maximumV3.h File Reference . . . . .	56
0.5.221 maximumV9.h File Reference . . . . .	56
0.5.222 maximumV9x4.h File Reference . . . . .	56
0.5.223 maxPool2x2s2hh.h File Reference . . . . .	57
0.5.224 maxPool3x3hh.h File Reference . . . . .	57
0.5.225 maxPool3x3s2hh.h File Reference . . . . .	57
0.5.226 maxTest3x3_fp16.h File Reference . . . . .	57
0.5.227 maxTest3x3_s16.h File Reference . . . . .	58
0.5.228 MDKdox-MvCv-intro.txt File Reference . . . . .	58
0.5.229 meanStdDev.h File Reference . . . . .	58
0.5.230 meshExpand.h File Reference . . . . .	58
0.5.231 meshGenerate.h File Reference . . . . .	59



0.5.232 minAggregate_line.h File Reference . . . . .	59
0.5.233 minKernel.h File Reference . . . . .	59
0.5.234 minKernel32.h File Reference . . . . .	59
0.5.235 minKernel64.h File Reference . . . . .	59
0.5.236 minMaxKernel.h File Reference . . . . .	59
0.5.237 minMaxPos.h File Reference . . . . .	60
0.5.238 minTest3x3_fp16.h File Reference . . . . .	60
0.5.239 minTest3x3_s16.h File Reference . . . . .	60
0.5.240 monoImbalance.h File Reference . . . . .	61
0.5.241 nonMax3x3_fp32.h File Reference . . . . .	61
0.5.242 nonMax3x3_u8.h File Reference . . . . .	61
0.5.243 opticalFlowPyrLK.h File Reference . . . . .	61
0.5.244 padKernel_u16.h File Reference . . . . .	62
0.5.245 padKernel_u8.h File Reference . . . . .	63
0.5.246 pixelPos.h File Reference . . . . .	64
0.5.247 pyrdown.h File Reference . . . . .	64
0.5.248 RANSAC.h File Reference . . . . .	64
0.5.249 rtems_config.h File Reference . . . . .	65
0.5.250 samplers.h File Reference . . . . .	65
0.5.251 scaleFp16.h File Reference . . . . .	66
0.5.252 scharr_fp16.h File Reference . . . . .	66
0.5.253 Sections.h File Reference . . . . .	66
0.5.254 Sections.h File Reference . . . . .	67
0.5.255 sLaplacian3x3.h File Reference . . . . .	69
0.5.256 sLaplacian3x3Fp16ToFp16.h File Reference . . . . .	69
0.5.257 sLaplacian5x5.h File Reference . . . . .	69
0.5.258 sLaplacian5x5Fp16ToFp16.h File Reference . . . . .	69
0.5.259 sLaplacian7x7.h File Reference . . . . .	69
0.5.260 sLaplacian7x7Fp16ToFp16.h File Reference . . . . .	69
0.5.261 smoothed_intensity.h File Reference . . . . .	69
0.5.262 sobel.h File Reference . . . . .	70
0.5.263 ssdPointLine7x7U8U32.h File Reference . . . . .	70
0.5.264 subPixelFilter.h File Reference . . . . .	70
0.5.265 sumOfAbsDiff11x11.h File Reference . . . . .	70
0.5.266 sumOfAbsDiff5x5.h File Reference . . . . .	70
0.5.267 sumOfSquaredDiff11x11.h File Reference . . . . .	70

0.5.268 sumOfSquaredDiff5x5.h File Reference . . . . .	70
0.5.269 sumOfSquaredDiff7x7U8ToU32.h File Reference . . . . .	70
0.5.270 svd.h File Reference . . . . .	71
0.5.271 thresholdBinaryRange.h File Reference . . . . .	71
0.5.272 thresholdBinaryU8.h File Reference . . . . .	71
0.5.273 thresholdDisparity.h File Reference . . . . .	71
0.5.274 thresholdDisparityFp16.h File Reference . . . . .	71
0.5.275 thresholdFilter.h File Reference . . . . .	71
0.5.276 thresholdFilterS16.h File Reference . . . . .	71
0.5.277 thresholdKernel.h File Reference . . . . .	71
0.5.278 tile_samplerRGB888.h File Reference . . . . .	72
0.5.279 tile_sampleru10.h File Reference . . . . .	72
0.5.280 tile_sampleru8.h File Reference . . . . .	72
0.5.281 warpMeshExpand.h File Reference . . . . .	72
0.5.282 warpMeshSample10bit.h File Reference . . . . .	72
0.5.283 warpMeshSample8bit.h File Reference . . . . .	73
0.5.284 warpMeshSample8bitStride.h File Reference . . . . .	73
0.5.285 warpMeshSampleFp16bit.h File Reference . . . . .	73
0.5.286 whiteBalanceBayerGBRG.h File Reference . . . . .	73
0.5.287 whiteBalanceRGB.h File Reference . . . . .	73

## **Index** **74**

### 0.1 Introduction

The document describes the kernel level interface to the Movidius Computer Vision Library

### 0.2 Data Structure Index

#### 0.2.1 Data Structures

Here are the data structures with brief descriptions:

bitbuffer_s . . . . .	15
fundamental_debug . . . . .	16
huffman_s . . . . .	16
mvcvPatternPoint . . . . .	17
RANSAC_Inputs . . . . .	18
RANSAC_Outputs . . . . .	18
ransac_params . . . . .	19

## 0.3 File Index

### 0.3.1 File List

Here is a list of all files with brief descriptions:

absoluteDiff.h	19
accumulateFp16.h	19
accumulateFp16MaxPool2s1.h	19
accumulateFp16MaxPool2s2.h	20
accumulateFp16MaxPool3s1.h	20
accumulateFp16MaxPool3s2.h	20
accumulateFp16MaxPool3s2withReLU.h	20
accumulateSquare.h	20
accumulateWeighted.h	20
addV2Fp16.h	20
aggregateAllPaths64.h	20
aggregateCostSGBM32_clamp.h	21
aggregateCostSGBM64.h	21
aggregateCostSGBM64_clamp.h	21
aggregateFivePaths32.h	21
aggregateFivePaths64.h	21
aggregateThreePaths32.h	21
aggregateThreePaths64.h	21
app_config.h	
Application configuration Leon header	22
arithmeticAdd.h	23
arithmeticAddmask.h	23
arithmeticSub.h	23
arithmeticSubFp16ToFp16.h	23
arithmeticSubmask.h	23
arithmeticSubU16.h	23
averageV3.h	23
avgPool7x7xk.h	24
bilateral5x5.h	24
bilinearInterpolation.h	24
bilinearInterpolation_u16.h	24
bilinearInterpolationAligned_u16.h	24
bitwiseAnd.h	24
bitwiseAndMask.h	24
bitwiseNot.h	25
bitwiseOr.h	25
bitwiseOrMask.h	25
bitwiseXor.h	25
bitwiseXorMask.h	25
boxFilter.h	25
boxFilter11x11.h	25
boxFilter13x13.h	25
boxFilter15x15.h	26
boxFilter3x3.h	26
boxFilter5x5.h	26
boxFilter7x7.h	26

boxFilter9x9.h	26
Buffers.h	26
calcBxBy.h	28
calcEpipolarDistance.h	28
calcG.h	28
canny.h	28
censusMatching16.h	28
censusMatching32.h	28
censusMatching64.h	28
censusMatching65.h	28
censusMatchingPyr.h	29
censusMatchingPyrOnePos.h	29
censusMatchingPyrOnePosWindow.h	29
censusMin16.h	29
censusMin3.h	29
censusMin32.h	29
censusMin64.h	29
censusMin65.h	30
censusMin7.h	30
censusMinConfidence32.h	30
censusMinConfidence64.h	31
censusMinConfidence64Subpixel.h	31
censusMinSubpixel3.h	32
censusMinSubpixel3Window.h	32
censusTransform11x11.h	32
censusTransform11x11u8.h	32
censusTransform5x5.h	32
censusTransformAverageRef7x7.h	32
censusTransformAverageRefMask7x7.h	32
channelExtract.h	32
computeAD32.h	33
computeAD64.h	33
computeADPyrOnePos.h	33
computeADPyrOnePosWindow.h	33
computeCombinedCost3.h	33
computeCombinedCost32.h	33
computeCombinedCost3Window.h	33
computeCombinedCost64.h	33
conv3x3fp32Schar.h	34
convert_fp16_u8.h	34
convert_u8_fp16.h	34
convertFp16ToQ4.h	34
convertFrom12BppTo8Bpp.h	34
convertYUV400ToYUV422.h	34
convolution11x11.h	34
convolution11x11s1hhhh.h	35
convolution11x11s1xxhx.h	35
convolution11x11s2hhhh.h	35
convolution11x11s2xxhx.h	35
convolution11x11s3hhhh.h	35
convolution11x11s4hhhh.h	35

convolution11x11s8hhhh.h	36
convolution15x1.h	36
convolution1x15.h	36
convolution1x5.h	36
convolution1x5Fp16ToFp16.h	36
convolution1x7.h	36
convolution1x7Fp16ToFp16.h	36
convolution1x9.h	36
convolution3x3.h	37
convolution3x3Fp16ToFp16.h	37
convolution3x3s2hhhh.h	37
convolution3x3s2xhhx.h	37
convolution3x3s3hhhh.h	37
convolution3x3s4hhhh.h	37
convolution3x3s8hhhh.h	37
convolution5x1.h	38
convolution5x1Fp16ToFp16.h	38
convolution5x5.h	38
convolution5x5Fp16ToFp16.h	38
convolution5x5s2hhhh.h	38
convolution5x5s3hhhh.h	38
convolution5x5s4hhhh.h	38
convolution5x5s8hhhh.h	38
convolution7x1.h	39
convolution7x1Fp16ToFp16.h	39
convolution7x7.h	39
convolution7x7Fp16ToFp16.h	39
convolution7x7Fp16ToU8.h	39
convolution7x7s2hhhh.h	39
convolution7x7s3hhhh.h	39
convolution7x7s4hhhh.h	39
convolution7x7s8hhhh.h	40
convolution9x1.h	40
convolution9x9.h	40
convolution9x9Fp16ToFp16.h	40
convolution9x9s2hhhh.h	40
convolution9x9s3hhhh.h	40
convolution9x9s4hhhh.h	40
convolution9x9s8hhhh.h	40
convSeparable11x11.h	41
convSeparable11x11Fp16ToFp16.h	41
convSeparable3x3.h	41
convSeparable3x3Fp16ToFp16.h	41
convSeparable5x5.h	41
convSeparable5x5Fp16ToFp16.h	41
convSeparable7x7.h	41
convSeparable7x7Fp16ToFp16.h	41
convSeparable9x9.h	42
convSeparable9x9Fp16ToFp16.h	42
cornerMinEigenVal.h	42
cornerMinEigenVal_fp32.h	42

cornerMinEigenVal_patched.h	42
cvtColorKernelChromaYUV420ToNV12.h	42
cvtColorKernelChromaYUV444ToNV12.h	42
cvtColorKernelRGBToYUV.h	43
cvtColorKernelRGBToYUV422.h	43
cvtColorKernelYUV422ToRGB.h	43
cvtColorKernelYUVToRGB.h	43
cvtColorNV21toRGB.h	43
cvtColorRGBfp16ToLumaU8.h	43
cvtColorRGBfp16ToUV420U8.h	43
cvtColorRGBtoChromaNV12.h	44
cvtColorRGBtoLuma.h	44
cvtColorRGBtoLumaNV12.h	44
cvtColorRGBtoNV12.h	44
cvtColorRGBtoNV21.h	44
cvtColorRGBtoUV.h	44
cvtColorRGBtoUV420.h	44
data.h	44
dct16bit.h	44
Defines.h	45
dilate.h	45
dilate3x3.h	45
dilate5x5.h	45
dilate7x7.h	46
disp2depth.h	46
downsampleBilinearLine.h	46
eigenValVec4x4Array.h	46
equalizeHist.h	46
erode3x3.h	46
erode5x5.h	46
erode7x7.h	46
f32UnitTestSample.h	47
fast9M2.h	47
fast9ScoreCv.h	47
fast9u16score.h	47
fastAtan2Positive.h	47
flipKernel.h	48
fm8Point.h	
Fundamental matrix computation interface for 8 Point algo	48
FundamentalMatrix.h	48
gauss.h	49
gauss1x5_u16in_u32out.h	49
gauss5x1_u32in_u16out.h	49
gaussHx2.h	49
gaussHx2_fp16.h	49
gaussVx2.h	49
gaussVx2_fp16.h	50
hammingDistance.h	50
harrisResponse.h	50
histogram.h	50
huffman_encode.h	50

integralImageSquareSumFloatM2.h	51
integralImageSquareSumM2.h	51
integralImageSumFloatM2.h	51
integralImageSumM2.h	51
integralImageSumU16U32.h	51
interpolatePixelBilinear.h	51
interpolatePixelBilinearS16.h	52
jpegGetBlockY420.h	52
kernels.h	52
kernelUnitTestSample.h	53
localMaxMin3x3_s16.h	54
lookupTable10to16.h	55
lookupTable10to8.h	55
lookupTable12to16.h	55
lookupTable12to8.h	55
lookupTable8to10.h	55
lookupTable8to8.h	55
matmulBT_xxii.h	56
matrixInv3x3_fp32.h	56
matrixInv4x4_fp32.h	56
matrixVectorMultfp16x4.h	56
maximumV2.h	56
maximumV3.h	56
maximumV9.h	56
maximumV9x4.h	56
maxPool2x2s2hh.h	57
maxPool3x3hh.h	57
maxPool3x3s2hh.h	57
maxTest3x3_fp16.h	57
maxTest3x3_s16.h	58
meanStdDev.h	58
meshExpand.h	58
meshGenerate.h	59
minAggregate_line.h	59
minKernel.h	59
minKernel32.h	59
minKernel64.h	59
minMaxKernel.h	59
minMaxPos.h	60
minTest3x3_fp16.h	60
minTest3x3_s16.h	60
monoImbalance.h	61
nonMax3x3_fp32.h	61
nonMax3x3_u8.h	61
opticalFlowPyrLK.h	
Optical Flow kernel Api	61
padKernel_u16.h	62
padKernel_u8.h	63
pixelPos.h	64
pyrdown.h	64
RANSAC.h	64

rtems_config.h	
RTEMS configuration Leon header	65
samplers.h	65
scaleFp16.h	66
scharr_fp16.h	66
opticalFlowPyrLK/unittest/dummy/shared/Sections.h	66
RANSAC/unittest/dummy/shared/Sections.h	67
sLaplacian3x3.h	69
sLaplacian3x3Fp16ToFp16.h	69
sLaplacian5x5.h	69
sLaplacian5x5Fp16ToFp16.h	69
sLaplacian7x7.h	69
sLaplacian7x7Fp16ToFp16.h	69
smoothed_intensity.h	69
sobel.h	70
ssdPointLine7x7U8U32.h	70
subPixelFilter.h	70
sumOfAbsDiff11x11.h	70
sumOfAbsDiff5x5.h	70
sumOfSquaredDiff11x11.h	70
sumOfSquaredDiff5x5.h	70
sumOfSquaredDiff7x7U8ToU32.h	70
svd.h	71
thresholdBinaryRange.h	71
thresholdBinaryU8.h	71
thresholdDisparity.h	71
thresholdDisparityFp16.h	71
thresholdFilter.h	71
thresholdFilterS16.h	71
thresholdKernel.h	71
tile_samplerRGB888.h	72
tile_sampleru10.h	72
tile_sampleru8.h	72
warpMeshExpand.h	72
warpMeshSample10bit.h	72
warpMeshSample8bit.h	73
warpMeshSample8bitStride.h	73
warpMeshSampleFp16bit.h	73
whiteBalanceBayerGBRG.h	73
whiteBalanceRGB.h	73

## 0.4 Data Structure Documentation

### 0.4.1 bitbuffer\_s Struct Reference

```
#include <huffman_encode.h>
```

#### Data Fields

- unsigned buf



- unsigned [n](#)

#### Field Documentation

unsigned bitbuffer\_s::buf

unsigned bitbuffer\_s::n

The documentation for this struct was generated from the following file:

- [huffman\\_encode.h](#)

### 0.4.2 fundamental\_debug Struct Reference

```
#include <RANSAC.h>
```

#### Data Fields

- int [RANSAC\\_support](#)
- int [RANSAC\\_nInliers](#)
- int [RANSAC\\_itterations](#)
- float [RANSAC\\_inliers\\_ratio](#)

#### Field Documentation

float fundamental\_debug::RANSAC\_inliers\_ratio

int fundamental\_debug::RANSAC\_itterations

int fundamental\_debug::RANSAC\_nInliers

int fundamental\_debug::RANSAC\_support

The documentation for this struct was generated from the following file:

- [RANSAC.h](#)

### 0.4.3 huffman\_s Struct Reference

```
#include <huffman_encode.h>
```

#### Data Fields

- const unsigned char(\* [haclen](#))[12]
- const unsigned short(\* [hacbit](#))[12]
- const unsigned char \* [hdclen](#)
- const unsigned short \* [hdcbit](#)
- const unsigned short \* [qtable](#)
- short [dc](#)

## Field Documentation

short `huffman_s::dc`

const unsigned short(\* `huffman_s::hacbit`)[12]

const unsigned char(\* `huffman_s::haclen`)[12]

const unsigned short\* `huffman_s::hdcbit`

const unsigned char\* `huffman_s::hdclen`

const unsigned short\* `huffman_s::qtable`

The documentation for this struct was generated from the following file:

- [huffman\\_encode.h](#)

## 0.4.4 mvcvPatternPoint Struct Reference

```
#include <smoothed_intensity.h>
```

## Data Fields

- float `x`
- float `y`
- float `sigma`

## Detailed Description

`mvcvSmoothed_intensity` kernel computes the average image intensity for N patches based on the integral image

Parameters

in	<i>kp_x</i>	- Patch center X coordinate
in	<i>kp_y</i>	- Patch center Y coordinate
in	<i>integral</i>	- pointer to integral image (u32)
in	<i>integral_stride</i>	- the stride of integral image lines in bytes
in	<i>pattern_pnt_lst</i>	- the list of the (x,y, sigma) values. X and Y means the offset from <i>kp_x</i> and <i>kp_y</i> , and sigma is the size of the patch
in	<i>pattern_pnt_sz</i>	- size of the <i>pattern_pnt_lst</i> . This is the number of patches
out	<i>intensities</i>	- Output values. The average intensities of the specified patches

## Returns

Nothing

## Field Documentation

float `mvcvPatternPoint::sigma`

```
float mvcvPatternPoint::x
```

```
float mvcvPatternPoint::y
```

The documentation for this struct was generated from the following file:

- [smoothed\\_intensity.h](#)

#### 0.4.5 RANSAC\_Inputs Struct Reference

```
#include <RANSAC.h>
```

##### Data Fields

- float \* [p1](#)
- float \* [p2](#)
- struct [ransac\\_params](#) \* [Params](#)

##### Field Documentation

```
float* RANSAC_Inputs::p1
```

```
float* RANSAC_Inputs::p2
```

```
struct ransac\_params* RANSAC_Inputs::Params
```

The documentation for this struct was generated from the following file:

- [RANSAC.h](#)

#### 0.4.6 RANSAC\_Outputs Struct Reference

```
#include <RANSAC.h>
```

##### Data Fields

- struct [fundamental\\_debug](#) \* [Debug](#)
- float \* [fm](#)
- int \* [inliers](#)

##### Field Documentation

```
struct fundamental\_debug* RANSAC_Outputs::Debug
```

```
float* RANSAC_Outputs::fm
```

```
int* RANSAC_Outputs::inliers
```

The documentation for this struct was generated from the following file:

- [RANSAC.h](#)

## 0.4.7 ransac\_params Struct Reference

```
#include <RANSAC.h>
```

### Data Fields

- float [inliers\\_ratio](#)
- float [confidence](#)
- float [dist\\_threshold](#)
- int [max\\_iterations](#)
- int [nPoints](#)

### Field Documentation

float [ransac\\_params::confidence](#)

float [ransac\\_params::dist\\_threshold](#)

float [ransac\\_params::inliers\\_ratio](#)

int [ransac\\_params::max\\_iterations](#)

int [ransac\\_params::nPoints](#)

The documentation for this struct was generated from the following file:

- [RANSAC.h](#)

## 0.5 File Documentation

### 0.5.1 absoluteDiff.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.2 accumulateFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.3 accumulateFp16MaxPool2s1.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.4 accumulateFp16MaxPool2s2.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.5 accumulateFp16MaxPool3s1.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.6 accumulateFp16MaxPool3s2.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.7 accumulateFp16MaxPool3s2withReLU.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.8 accumulateSquare.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.9 accumulateWeighted.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.10 addV2Fp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.11 aggregateAllPaths64.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Macros

- #define [DISPARITIES](#) (64)

## Macro Definition Documentation

#define [DISPARITIES](#) (64)

### 0.5.12 [aggregateCostSGBM32\\_clamp.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.13 [aggregateCostSGBM64.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.14 [aggregateCostSGBM64\\_clamp.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.15 [aggregateFivePaths32.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.16 [aggregateFivePaths64.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.17 [aggregateThreePaths32.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.18 [aggregateThreePaths64.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.19 app\_config.h File Reference

Application configuration Leon header.

#### Macros

- #define [APP\\_MSS\\_CLOCKS](#)
- #define [APP\\_UPA\\_CLOCKS](#)

#### Functions

- int [initClocksAndMemory](#) (void)

#### Detailed Description

Application configuration Leon header.

#### Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

#### Macro Definition Documentation

#define [APP\\_MSS\\_CLOCKS](#)

##### Value:

```
(DEV_MSS_APB_SLV | \
  DEV_MSS_APB2_CTRL | \
  DEV_MSS_AXI_BRIDGE | \
  DEV_MSS_MXI_CTRL | \
  DEV_MSS_MXI_DEFSLV )
```

#define [APP\\_UPA\\_CLOCKS](#)

##### Value:

```
(DEV_UPA_SH0 | \
  DEV_UPA_SHAVE_L2 | \
  DEV_UPA_CDMA | \
  DEV_UPA_CTRL )
```

#### Function Documentation

int [initClocksAndMemory](#) ( void )

Setup all the clock configurations needed by this application and also the ddr

## Returns

0 on success, non-zero otherwise

### 0.5.20 [arithmeticAdd.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.21 [arithmeticAddmask.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.22 [arithmeticSub.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.23 [arithmeticSubFp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.24 [arithmeticSubmask.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.25 [arithmeticSubU16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.26 [averageV3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```



### 0.5.27 avgPool7x7xk.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.28 bilateral5x5.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.29 bilinearInterpolation.h File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

### 0.5.30 bilinearInterpolation\_u16.h File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

### 0.5.31 bilinearInterpolationAligned\_u16.h File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

### 0.5.32 bitwiseAnd.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.33 bitwiseAndMask.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.34 bitwiseNot.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.35 bitwiseOr.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.36 bitwiseOrMask.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.37 bitwiseXor.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.38 bitwiseXorMask.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.39 boxFilter.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.40 boxFilter11x11.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.41 boxFilter13x13.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.42 boxFilter15x15.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.43 boxFilter3x3.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.44 boxFilter5x5.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.45 boxFilter7x7.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.46 boxFilter9x9.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.47 Buffers.h File Reference

```
#include <mv_types.h>
#include <Sections.h>
#include <Defines.h>
```

#### Macros

- `#define DDR\_BUFF DDR\_DATA`

#### Variables

- `u8 DDR\_BUFF inputFrameA [(MAX_FRAME_WIDTH+SEARCH_WIN_SZ *2)*MAX_FRAME_HEIGHT]`
- `u8 DDR\_BUFF pyrAL1 [(MAX_FRAME_WIDTH/2+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/2)]`

- `u8 DDR_BUFF pyrAL2 [(MAX_FRAME_WIDTH/4+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/4)]`
- `u8 DDR_BUFF pyrAL3 [(MAX_FRAME_WIDTH/8+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/8)]`
- `u8 DDR_BUFF pyrAL4 [(MAX_FRAME_WIDTH/16+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/16)]`
- `u8 DDR_BUFF inputFrameB [(MAX_FRAME_WIDTH+SEARCH_WIN_SZ *2)*MAX_FRAME_HEIGHT]`
- `u8 DDR_BUFF pyrBL1 [(MAX_FRAME_WIDTH/2+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/2)]`
- `u8 DDR_BUFF pyrBL2 [(MAX_FRAME_WIDTH/4+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/4)]`
- `u8 DDR_BUFF pyrBL3 [(MAX_FRAME_WIDTH/8+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/8)]`
- `u8 DDR_BUFF pyrBL4 [(MAX_FRAME_WIDTH/16+SEARCH_WIN_SZ *2)*(MAX_FRAME_HEIGHT/16)]`

## Macro Definition Documentation

```
#define DDR_BUFF DDR_DATA
```

## Variable Documentation

```
u8 DDR_BUFF inputFrameA[(MAX_FRAME_WIDTH+SEARCH_WIN_SZ
*2)*MAX_FRAME_HEIGHT]
```

```
u8 DDR_BUFF inputFrameB[(MAX_FRAME_WIDTH+SEARCH_WIN_SZ
*2)*MAX_FRAME_HEIGHT]
```

```
u8 DDR_BUFF pyrAL1[(MAX_FRAME_WIDTH/2+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/2)]
```

```
u8 DDR_BUFF pyrAL2[(MAX_FRAME_WIDTH/4+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/4)]
```

```
u8 DDR_BUFF pyrAL3[(MAX_FRAME_WIDTH/8+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/8)]
```

```
u8 DDR_BUFF pyrAL4[(MAX_FRAME_WIDTH/16+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/16)]
```

```
u8 DDR_BUFF pyrBL1[(MAX_FRAME_WIDTH/2+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/2)]
```

```
u8 DDR_BUFF pyrBL2[(MAX_FRAME_WIDTH/4+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/4)]
```

```
u8 DDR_BUFF pyrBL3[(MAX_FRAME_WIDTH/8+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/8)]
```

```
u8 DDR_BUFF pyrBL4[(MAX_FRAME_WIDTH/16+SEARCH_WIN_SZ
*2)*(MAX_FRAME_HEIGHT/16)]
```

#### 0.5.48 `calcBxBy.h` File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.49 `calcEpipolarDistance.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.50 `calcG.h` File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.51 `canny.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.52 `censusMatching16.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.53 `censusMatching32.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.54 `censusMatching64.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.55 `censusMatching65.h` File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

#### 0.5.56 censusMatchingPyr.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.57 censusMatchingPyrOnePos.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.58 censusMatchingPyrOnePosWindow.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.59 censusMin16.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.60 censusMin3.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.61 censusMin32.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.62 censusMin64.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

### 0.5.63 censusMin65.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.64 censusMin7.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Macros

- #define [DISPARITIES](#) 7

## Macro Definition Documentation

#define [DISPARITIES](#) 7

**mvcvCensusMin** - computes minimum of 7 disparity costs values

Parameters

<i>in</i>	<i>in</i>	- pointer to disparity costs
<i>out</i>	<i>out</i>	- array of disparity cost
<i>in</i>	<i>width</i>	- width of the input lines

Returns

Nothing

### 0.5.65 censusMinConfidence32.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Functions

- [MVCV\\_FUNC](#) (void, **mvcvCensusMinConfidence32**, u8 \*input, u8 \*minimumPosition, u8 \*minimumValue, u32 [width](#))

## Function Documentation

**MVCV\_FUNC** ( void , **mvcvCensusMinConfidence32** , u8 \* input, u8 \* minimumPosition, u8 \* minimumValue, u32 width )

**mvcvCensusMinConfidence32**

## Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of minimum positions
out	<i>outm</i>	- array of computation: $\text{scaleFactor} * \text{minCost} / \text{secondMinCost}$ between minimum value and second minimum
in	<i>width</i>	- width of the input lines

## Returns

Nothing

## 0.5.66 censusMinConfidence64.h File Reference

```
#include <mv_types.h>
#include <mv_cv_macro.h>
```

## Functions

- **MVCV\_FUNC** (void, mvCensusMinConfidence64, u8 \*input, u8 \*minimumPosition, u8 \*minimumValue, u32 width)

## Function Documentation

**MVCV\_FUNC** ( void , mvCensusMinConfidence64 , u8 \* input, u8 \* minimumPosition, u8 \* minimumValue, u32 width )

mvCensusMinConfidence64

## Parameters

in	<i>in</i>	- pointer to disparity costs
out	<i>out</i>	- array of minimum positions
out	<i>outm</i>	- array of computation: $\text{scaleFactor} * \text{minCost} / \text{secondMinCost}$ between minimum value and second minimum
in	<i>width</i>	- width of the input lines

## Returns

Nothing

## 0.5.67 censusMinConfidence64Subpixel.h File Reference

```
#include <mv_types.h>
#include <mv_cv_macro.h>
```



### 0.5.68 [censusMinSubpixel3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.69 [censusMinSubpixel3Window.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.70 [censusTransform11x11.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.71 [censusTransform11x11u8.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.72 [censusTransform5x5.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.73 [censusTransformAverageRef7x7.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.74 [censusTransformAverageRefMask7x7.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.75 [channelExtract.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.76 [computeAD32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.77 [computeAD64.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.78 [computeADPyrOnePos.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.79 [computeADPyrOnePosWindow.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.80 [computeCombinedCost3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.81 [computeCombinedCost32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.82 [computeCombinedCost3Window.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.83 [computeCombinedCost64.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.84 [conv3x3fp32Schar.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.85 [convert\\_fp16\\_u8.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.86 [convert\\_u8\\_fp16.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.87 [convertFp16ToQ4.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.88 [convertFrom12BppTo8Bpp.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.89 [convertYUV400ToYUV422.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.90 [convolution11x11.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.91 convolution1x1s1hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.92 convolution1x1s1xxhx.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.93 convolution1x1s2hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.94 convolution1x1s2xxhx.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.95 convolution1x1s3hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.96 convolution1x1s4hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.97 convolution1x1s8hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
#include <stdio.h>
#include <stdlib.h>
```

### 0.5.98 convolution15x1.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.99 convolution1x15.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.100 convolution1x5.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.101 convolution1x5Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.102 convolution1x7.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.103 convolution1x7Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.104 convolution1x9.h File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

#### 0.5.105 convolution3x3.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.106 convolution3x3Fp16ToFp16.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.107 convolution3x3s2hhhh.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.108 convolution3x3s2xhhx.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.109 convolution3x3s3hhhh.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.110 convolution3x3s4hhhh.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.111 convolution3x3s8hhhh.h File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.112 convolution5x1.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.113 convolution5x1Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.114 convolution5x5.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.115 convolution5x5Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.116 convolution5x5s2hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.117 convolution5x5s3hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.118 convolution5x5s4hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.119 convolution5x5s8hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.120 convolution7x1.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.121 convolution7x1Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.122 convolution7x7.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.123 convolution7x7Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.124 convolution7x7Fp16ToU8.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.125 convolution7x7s2hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.126 convolution7x7s3hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.127 convolution7x7s4hhhh.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```



#### 0.5.128 [convolution7x7s8hhhh.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.129 [convolution9x1.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.130 [convolution9x9.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.131 [convolution9x9Fp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.132 [convolution9x9s2hhhh.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.133 [convolution9x9s3hhhh.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.134 [convolution9x9s4hhhh.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.135 [convolution9x9s8hhhh.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.136 [convSeparable11x11.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.137 [convSeparable11x11Fp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.138 [convSeparable3x3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.139 [convSeparable3x3Fp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.140 [convSeparable5x5.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.141 [convSeparable5x5Fp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.142 [convSeparable7x7.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.143 [convSeparable7x7Fp16ToFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.144 [convSeparable9x9.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.145 [convSeparable9x9Fp16ToFp16.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.146 [cornerMinEigenVal.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.147 [cornerMinEigenVal\\_fp32.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.148 [cornerMinEigenVal\\_patched.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.149 [cvtColorKernelChromaYUV420ToNV12.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.150 [cvtColorKernelChromaYUV444ToNV12.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### Variables

- u8 \* [inV](#) [2]
- u8 u8 \* [outUV](#)
- u8 u8 u32 [width](#)

## Variable Documentation

u8\* inV[2]

u8 u8\* outUV

u8 u8 u32 width

### 0.5.151 [cvtColorKernelRGBToYUV.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.152 [cvtColorKernelRGBToYUV422.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.153 [cvtColorKernelYUV422ToRGB.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.154 [cvtColorKernelYUVToRGB.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.155 [cvtColorNV21toRGB.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.156 [cvtColorRGBfp16ToLumaU8.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.157 [cvtColorRGBfp16ToUV420U8.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.158 cvtColorRGBtoChromaNV12.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.159 cvtColorRGBtoLuma.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.160 cvtColorRGBtoLumaNV12.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.161 cvtColorRGBtoNV12.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.162 cvtColorRGBtoNV21.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.163 cvtColorRGBtoUV.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.164 cvtColorRGBtoUV420.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### [0.5.165 data.h File Reference](#)

#### [0.5.166 dct16bit.h File Reference](#)

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

## 0.5.167 Defines.h File Reference

### Macros

- #define [MAX\\_FRAME\\_WIDTH](#) (1280)
- #define [MAX\\_FRAME\\_HEIGHT](#) (720)
- #define [MAX\\_FEATURES](#) (300)
- #define [MAX\\_PYRAMID\\_LEVELS](#) (4)
- #define [SEARCH\\_WIN\\_SZ](#) (5)
- #define [STOP\\_CRIT\\_MAX\\_ITER](#) (9)
- #define [STOP\\_CRIT\\_EPS](#) (0.01f)
- #define [HEAP\\_SIZE](#) (20 \* 1024)

### Macro Definition Documentation

```
#define HEAP_SIZE (20 * 1024)
```

```
#define MAX_FEATURES (300)
```

```
#define MAX_FRAME_HEIGHT (720)
```

```
#define MAX_FRAME_WIDTH (1280)
```

```
#define MAX_PYRAMID_LEVELS (4)
```

```
#define SEARCH_WIN_SZ (5)
```

```
#define STOP_CRIT_EPS (0.01f)
```

```
#define STOP_CRIT_MAX_ITER (9)
```

## 0.5.168 dilate.h File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

## 0.5.169 dilate3x3.h File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

## 0.5.170 dilate5x5.h File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

#### 0.5.171 [dilate7x7.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.172 [disp2depth.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.173 [downsampleBilinearLine.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.174 [eigenValVec4x4Array.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.175 [equalizeHist.h File Reference](#)

```
#include <mvcv_macro.h>
#include <mv_types.h>
```

#### 0.5.176 [erode3x3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.177 [erode5x5.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.178 [erode7x7.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.179 f32UnitTestSample.h File Reference

```
#include <mv_types.h>
```

### Functions

- void [mvuDotProductStreaming](#) (fp32 \*vec\_a, fp32 \*\*vec\_b\_list, fp32 \*out, u32 num\_vectors)
- void [dotProductStreaming](#) (fp32 \*vec\_a, fp32 \*\*vec\_b\_list, fp32 \*out, u32 num\_vectors)

### Function Documentation

```
void dotProductStreaming ( fp32 * vec_a, fp32 ** vec_b_list, fp32 * out, u32 num_vectors )
```

```
void mvuDotProductStreaming ( fp32 * vec_a, fp32 ** vec_b_list, fp32 * out, u32 num_vectors )
```

This kernel computes the dot product between vec\_a and each vector from vec\_b\_list (it uses the mvuDot built-in function from Vector Utility Library)

#### Parameters

in	<i>vec_a</i>	- Array of floats
in	<i>vec_b_list</i>	- Array of pointers to floats
out	<i>out</i>	- Array of floats for dot product results
in	<i>num_vectors</i>	- Number of vectors for vec_b_list

## 0.5.180 fast9M2.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.181 fast9ScoreCv.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.182 fast9u16score.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.183 fastAtan2Positive.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```



## 0.5.184 flipKernel.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.185 fm8Point.h File Reference

Fundamental matrix computation interface for 8 Point algo.

### Functions

- void [fm8Point](#) (float \*p\_1, float \*p\_2, float \*fm)  
: Compute fundamental matrix using 8-Point algo
- void [halffm8Point](#) (float \*p\_1, float \*p\_2, float \*fm)

### Detailed Description

Fundamental matrix computation interface for 8 Point algo.

### Copyright

All code copyright Movidius Ltd 2015, all rights reserved. For License Warranty see: common/license.txt

### Function Documentation

[void fm8Point](#) ( float \* p\_1, float \* p\_2, float \* fm )

: Compute fundamental matrix using 8-Point algo

#### Parameters

in	$p_1$	- corners from the first image
in	$p_2$	- corresponding points in the second image
out	$fm$	- fundamental matrix

[void halffm8Point](#) ( float \* p\_1, float \* p\_2, float \* fm )

## 0.5.186 FundamentalMatrix.h File Reference

```
#include "RANSAC.h"
```

### Macros

- #define [POINTS\\_PER\\_SAMPLE](#) 8

## Functions

- void [findFundamentalMat](#) (float \*p\_1, float \*p\_2, int nPoints, float \*fm, int \*inliers, struct [fundamental\\_debug](#) \*Debug, float RANSAC\_inliers\_ratio=0.5, float RANSAC\_confidence=0.999, float RANSAC\_dist\_threshold=1, int RANSAC\_max\_iterations=100)

## Macro Definition Documentation

```
#define POINTS_PER_SAMPLE 8
```

## Function Documentation

```
void findFundamentalMat ( float * p_1, float * p_2, int nPoints, float * fm, int * inliers, struct
fundamental_debug * Debug, float RANSAC_inliers_ratio = 0.5, float RANSAC_confidence =
0.999, float RANSAC_dist_threshold = 1, int RANSAC_max_iterations = 100 )
```

### 0.5.187 [gauss.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.188 [gauss1x5\\_u16in\\_u32out.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.189 [gauss5x1\\_u32in\\_u16out.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.190 [gaussHx2.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.191 [gaussHx2\\_fp16.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.192 [gaussVx2.h](#) File Reference

```
#include <mv_types.h>
```

```
#include <mvcv_macro.h>
```

#### 0.5.193 `gaussVx2_fp16.h` File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.194 `hammingDistance.h` File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.195 `harrisResponse.h` File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.196 `histogram.h` File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### 0.5.197 `huffman_encode.h` File Reference

```
#include <mv_types.h>  
#include <mvcv_macro.h>
```

#### Data Structures

- struct `huffman_s`
- struct `bitbuffer_s`

#### Typedefs

- typedef struct `huffman_s` `huffman_t`
- typedef struct `bitbuffer_s` `bitbuffer_t`

#### Typedef Documentation

```
typedef struct bitbuffer_s bitbuffer_t
```

```
typedef struct huffman_s huffman_t
```

#### 0.5.198 [integralImageSquareSumFloatM2.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.199 [integralImageSquareSumM2.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.200 [integralImageSumFloatM2.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.201 [integralImageSumM2.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.202 [integralImageSumU16U32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.203 [interpolatePixelBilinear.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### Functions

- [MVCV\\_FUNC](#) (half, mvcvInterpolatePixelBilinear, half \*line1, half \*line2, fp32 x, fp32 y)

### Function Documentation

[MVCV\\_FUNC](#) ( half , mvcvInterpolatePixelBilinear , half \* line1, half \* line2, fp32 x, fp32 y )

Bilinear interpolation of four pixels

## Parameters

in	<i>Line1</i>	- First input line
in	<i>Line2</i>	- Second input line
in	<i>x</i>	- The x coordinate of the pixel.
in	<i>y</i>	- The y coordinate of the pixel. Because we work with two lines only, only the fractional part of the number matters.

## Returns

The value of the interpolated pixel.

## 0.5.204 interpolatePixelBilinearS16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Functions

- [MVCV\\_FUNC](#) (float, mvcvInterpolatePixelBilinearS16, s16 \*line1, s16 \*line2, fp32 x, fp32 y)

## Function Documentation

[MVCV\\_FUNC](#) ( float , mvcvInterpolatePixelBilinearS16 , s16 \* line1, s16 \* line2, fp32 x, fp32 y )

Bilinear interpolation of four pixels

## Parameters

in	<i>Line1</i>	- First input line
in	<i>Line2</i>	- Second input line
in	<i>x</i>	- The x coordinate of the pixel.
in	<i>y</i>	- The y coordinate of the pixel. Because we work with two lines only, only the fractional part of the number matters.

## Returns

The value of the interpolated pixel.

## 0.5.205 jpegGetBlockY420.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.206 kernels.h File Reference

```
#include <mvcv_types.h>
```

```
#include "calcG.h"
#include "calcBxBy.h"
#include "bilinearInterpolation_u16.h"
#include "bilinearInterpolationAligned_u16.h"
#include "conv3x3fp32Scharr.h"
```

## 0.5.207 kernelUnitTestSample.h File Reference

```
#include <mv_types.h>
```

### Functions

- `u8 * dummy_kernel_result` (`u8 *in_tile`, `u8 *out_tile_space`, `int width`, `int height`)
- `void blendkernel` (`u8 *in1`, `u8 *in2`, `u8 *out`, `u32 width`, `u32 height`)
- `void avg3x5kernel_newint` (`u8 **in_lines`, `u8 **out_lines`, `u32 width`)
- `void avg1x5kernel` (`u8 **in`, `u8 **out`, `u32 width`)

### Function Documentation

`void avg1x5kernel ( u8 ** in, u8 ** out, u32 width )`

Kernel that blurs with 1x5 radius (5 lines, 1 column)

Parameters

<code>in</code>	-	In address to plane 1 to blend
<code>out</code>	-	Out pointer to output memory with blended result
<code>in</code>	-	Width width of the planes

Returns

none

`void avg3x5kernel_newint ( u8 ** in_lines, u8 ** out_lines, u32 width )`

Kernel that blurs with 3x5 radius (5 lines, 3 columns)

Parameters

<code>in</code>	-	In_lines pointer to the array of input line pointers
<code>out</code>	-	Out_lines pointer to the array of output line pointers
<code>in</code>	-	Width width of the planes

Returns

none

`void blendkernel ( u8 * in1, u8 * in2, u8 * out, u32 width, u32 height )`

Kernel that bit or blends two planes

## Parameters

in	-	In1 address to plane 1 to blend
in	-	In2 address to plane 2 to blend
out	-	Out pointer to output memory with blended result
in	-	Width width of the planes
in	-	Height height of the planes

## Returns

none

`u8* dummy_kernel_result ( u8 * in_tile, u8 * out_tile_space, int width, int height )`

Dummy kernel that only copies data around

## Parameters

in	-	In_tile address of the memory where the input tile is stored
in	-	Out_tile_space address of the memory where the output tile is stored
in	-	Width width of the tile
in	-	Height height of the tile

## Returns

`u8*` pointer to the memory area where the tile was copied

## 0.5.208 localMaxMin3x3\_s16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Functions

- **MVCV\_FUNC** (void, mvcvLocalMaxMin3x3\_s16, s16 \*\*inBuffer, u32 candidateLocationIn[], u32 countIn, u32 minLocationList[], u32 maxLocationList[], u32 \*minCount, u32 \*maxCount)

## Function Documentation

`MVCV_FUNC ( void , mvcvLocalMaxMin3x3_s16 , s16 ** inBuffer, u32 candidateLocationIn[], u32 countIn, u32 minLocationList[], u32 maxLocationList[], u32 * minCount, u32 * maxCount )`

This kernel will find the points which are minimums or maximums in their 3x3 zone. The points from the middle line are compared to their neighbors

## Parameters

---

in	<i>inBuffer</i>	- Address of the s16 image buffer. This buffer should have a size of 3 * width * sizeof(s16) bytes (3 lines)
in	<i>candidate-LocationIn</i>	- the X coordinates of the candidates
in	<i>countIn</i>	- number of candidates
out	<i>minLocation-List</i>	- u32 buffer, where the function will save the X coordinates of the minimums found
out	<i>maxLocation-List</i>	- u32 buffer, where the function will save the X coordinates of the maximums found
out	<i>minCount</i>	- The function will save the number of minimums found to this address
out	<i>maxCount</i>	- The function will save the number of maximums found to this address

### 0.5.209 [lookupTable10to16.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.210 [lookupTable10to8.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.211 [lookupTable12to16.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.212 [lookupTable12to8.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.213 [lookupTable8to10.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.214 [lookupTable8to8.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```



#### 0.5.215 [matmulBT\\_xxii.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.216 [matrixInv3x3\\_fp32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.217 [matrixInv4x4\\_fp32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.218 [matrixVectorMultfp16x4.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.219 [maximumV2.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.220 [maximumV3.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.221 [maximumV9.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.222 [maximumV9x4.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.223 `maxPool2x2s2hh.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.224 `maxPool3x3hh.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.225 `maxPool3x3s2hh.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.226 `maxTest3x3_fp16.h` File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Functions

- **MVCV\_FUNC** (void, `mvcvMaxTest3x3_fp16`, half \*`inBufferCandidates`, half \*\*`inBuffer`, u32 `width`, u32 `maxLocationsIn[]`, u32 `maxLocationsOut[]`, u32 `maxCountIn`, u32 \*`maxCountOut`)

## Function Documentation

`MVCV_FUNC` ( void , `mvcvMaxTest3x3_fp16` , half \* `inBufferCandidates`, half \*\* `inBuffer`, u32 `width`, u32 `maxLocationsIn[]`, u32 `maxLocationsOut[]`, u32 `maxCountIn`, u32 \* `maxCountOut` )

This kernel will compare the points from `inBufferCandidates` to the corresponding 3x3 zone of `inBuffer`. The function will check `maxCountIn` locations.

Parameters

in	<i>inBufferCandidates</i>	- Fp16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- Address of the fp16 image buffer. This buffer should have a size of 3 * width * sizeof(fp16) bytes (3 lines)
in	<i>width</i>	- Line width in pixels as u32
in	<i>maxLocationsIn</i>	- Gives the x coordinates of the candidates. Only these candidates are checked.

out	<i>maxLocationsOut</i>	- The values from maxLocationsIn which passed the filter.
in	<i>maxCountIn</i>	- Number of values in maxLocationsIn buffer
out	<i>maxCountOut</i>	- Number of values in maxLocationsOut buffer

### 0.5.227 maxTest3x3\_s16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### Functions

- **MVCV\_FUNC** (void, mvcvMaxTest3x3\_s16, s16 \*inBufferCandidates, s16 \*\*inBuffer, u32 maxLocationsIn[], u32 maxLocationsOut[], u32 maxCountIn, u32 \*maxCountOut)

#### Function Documentation

**MVCV\_FUNC** ( void , mvcvMaxTest3x3\_s16 , s16 \* inBufferCandidates, s16 \*\* inBuffer, u32 maxLocationsIn[], u32 maxLocationsOut[], u32 maxCountIn, u32 \* maxCountOut )

This kernel will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer. The function will check maxCountIn locations.

#### Parameters

in	<i>inBufferCandidates</i>	- s16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- pointers to the u16 input line. It will contain three pointers for three lines
in	<i>maxLocationsIn</i>	- Gives the x coordinates of the candidates. Only these candidates are checked.
out	<i>maxLocationsOut</i>	- The values from maxLocationsIn which passed the filter.
in	<i>maxCountIn</i>	- Number of values in maxLocationsIn buffer
out	<i>maxCountOut</i>	- Number of values in maxLocationsOut buffer

### 0.5.228 MDKdox-MvCv-intro.txt File Reference

### 0.5.229 meanStdDev.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.230 meshExpand.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Functions

- [MVCV\\_FUNC](#) (void, mvcvMeshExpand, u8 \*out, u8 \*\*in, u32 width, half \*\*mesh, float my)

## Function Documentation

[MVCV\\_FUNC](#) ( void , mvcvMeshExpand , u8 \* out, u8 \*\* in, u32 width, half \*\* mesh, float my )

### 0.5.231 [meshGenerate.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.232 [minAggregate\\_line.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.233 [minKernel.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.234 [minKernel32.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.235 [minKernel64.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.236 [minMaxKernel.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.237 minMaxPos.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.238 minTest3x3\_fp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### Functions

- **MVCV\_FUNC** (void, mvcvMinTest3x3\_fp16, half \*inBufferCandidates, half \*\*inBuffer, u32 width, u32 minLocationsIn[], u32 minLocationsOut[], u32 minCountIn, u32 \*minCountOut)

#### Function Documentation

**MVCV\_FUNC** ( void , mvcvMinTest3x3\_fp16 , half \* inBufferCandidates, half \*\* inBuffer, u32 width, u32 minLocationsIn[], u32 minLocationsOut[], u32 minCountIn, u32 \* minCountOut )

This function will compare the points from inBufferCandidates to the corresponding 3x3 zone of inBuffer. The function will check minCountIn locations.

Parameters

in	<i>inBufferCandidates</i>	- fp16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- address of the fp16 image buffer. This buffer should have a size of 3 * width * sizeof(fp16) bytes (3 lines)
in	<i>width</i>	- Line width in pixels as u32
in	<i>minLocationsIn</i>	- gives the x coordinates of the candidates. Only these candidates are checked.
out	<i>minLocationsOut</i>	- the values from minLocationsIn which passed the filter.
in	<i>minCountIn</i>	- number of values in minLocationsIn buffer
out	<i>minCountOut</i>	- number of values in minLocationsOut buffer

### 0.5.239 minTest3x3\_s16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### Functions

- **MVCV\_FUNC** (void, mvcvMinTest3x3\_s16, s16 \*inBufferCandidates, s16 \*\*inBuffer, u32 minLocationsIn[], u32 minLocationsOut[], u32 minCountIn, u32 \*minCountOut)

## Function Documentation

`MVCV_FUNC ( void , mvcvMinTest3x3_s16 , s16 * inBufferCandidates, s16 ** inBuffer, u32 minLocationsIn[], u32 minLocationsOut[], u32 minCountIn, u32 * minCountOut )`

This kernel will compare the points from `inBufferCandidates` to the corresponding 3x3 zone of `inBuffer`. The function will check `minCountIn` locations.

Parameters

in	<i>inBuffer-Candidates</i>	- s16 buffer, where the candidates can be found
in	<i>inBuffer</i>	- pointers to the u16 input line. It will contain three pointers for three lines
in	<i>minLocations-In</i>	- Gives the x coordinates of the candidates. Only these candidates are checked.
out	<i>minLocations-Out</i>	- The values from <code>minLocationsIn</code> which passed the filter.
in	<i>minCountIn</i>	- Number of values in <code>minLocationsIn</code> buffer
out	<i>minCountOut</i>	- Number of values in <code>minLocationsOut</code> buffer

### 0.5.240 monoImbalance.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.241 nonMax3x3\_fp32.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.242 nonMax3x3\_u8.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.243 opticalFlowPyrLK.h File Reference

Optical Flow kernel Api.

```
#include <mvcv_types.h>
#include <mvcv_macro.h>
```

## Functions

- s32 [calcOpticalFlowPyrLK](#) (u8 \*imgA, u8 \*imgB, mvCvSize size, u8 \*pyrA, u8 \*pyrB, mvCvPoint2D32f \*featuresA, mvCvPoint2D32f \*featuresB, u8 \*status, fp32 \*error, mvCvSize winSize, u32 level, mvCvTermCriteria criteria, u32 flags, u32 nb\_points)

## Detailed Description

Optical Flow kernel Api.

## Copyright

All code copyright Movidius Ltd 2012, all rights reserved For License Warranty see- : common/license.txt

## Function Documentation

s32 [calcOpticalFlowPyrLK](#) ( u8 \* imgA, u8 \* imgB, mvCvSize size, u8 \* pyrA, u8 \* pyrB, mvCvPoint2D32f \* featuresA, mvCvPoint2D32f \* featuresB, u8 \* status, fp32 \* error, mvCvSize winSize, u32 level, mvCvTermCriteria criteria, u32 flags, u32 nb\_points )

Optical Flow algorithm - golden implementation extracted from OpenCV

## Parameters

in	<i>imgA</i>	First input image
in	<i>imgB</i>	Second input image
in	<i>size</i>	Size of the input image
out	<i>pyrA</i>	Pointer to a preallocated buffer which needs to accomodate all pyramid levels
out	<i>pyrB</i>	Pointer to a preallocated buffer which needs to accomodate all pyramid levels
in	<i>featuresA</i>	Input list of features
out	<i>featuresB</i>	Output list of tracked features
out	<i>status</i>	Found/not found status of each input feature
out	<i>error</i>	Error value for each tracked feature
in	<i>winSize</i>	Size of the search window used for tracking
in	<i>level</i>	Number of pyramid levels to generate
in	<i>criteria</i>	Termination criteria for the tracking algorithm (minimum error or maximum iterations reached)
in	<i>flags</i>	Flags specifying various runtime options
in	<i>nb_points</i>	Number of input features

## Returns

Returns 0 if it has finished succesfully, non-0 for error

## 0.5.244 padKernel\_u16.h File Reference

```
#include <mv_types.h>
#include <mv_cv_macro.h>
```

## Enumerations

- enum { **Left** = 0, **Right** = 1, **LeftAndRight** = 2 }
- enum {  
**AllZero** = 0, **AllOne** = 1, **Mirror** = 2, **BlackPixel** = 3,  
**WhitePixel** = 4, **PixelValue** = 5 }

## Enumeration Type Documentation

### anonymous enum

Enumerator

***Left***  
***Right***  
***LeftAndRight***

### anonymous enum

Enumerator

***AllZero***  
***AllOne***  
***Mirror***  
***BlackPixel***  
***WhitePixel***  
***PixelValue***

## 0.5.245 padKernel\_u8.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Enumerations

- enum { **Left** = 0, **Right** = 1, **LeftAndRight** = 2 }
- enum {  
**AllZero** = 0, **AllOne** = 1, **Mirror** = 2, **BlackPixel** = 3,  
**WhitePixel** = 4, **PixelValue** = 5 }

## Enumeration Type Documentation

### anonymous enum

Enumerator

***Left***  
***Right***  
***LeftAndRight***



anonymous enum

Enumerator

***AllZero***

***AllOne***

***Mirror***

***BlackPixel***

***WhitePixel***

***PixelValue***

## 0.5.246 pixelPos.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.247 pyrdown.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## 0.5.248 RANSAC.h File Reference

Data Structures

- struct [ransac\\_params](#)
- struct [fundamental\\_debug](#)
- struct [RANSAC\\_Inputs](#)
- struct [RANSAC\\_Outputs](#)

Macros

- #define [MAX\\_NUMPOINTS](#) 1000

Typedefs

- typedef struct [RANSAC\\_Inputs](#) [RANSAC\\_input](#)
- typedef struct [RANSAC\\_Outputs](#) [RANSAC\\_output](#)

Macro Definition Documentation

```
#define MAX\_NUMPOINTS 1000
```

Typedef Documentation

```
typedef struct RANSAC_Inputs RANSAC_input
```

```
typedef struct RANSAC_Outputs RANSAC_output
```

#### 0.5.249 `rtems_config.h` File Reference

RTEMS configuration Leon header.

```
#include "app_config.h"
```

#### Functions

- [BSP\\_SET\\_CLOCK](#) (12000, 500000, 1, 1, DEFAULT RTEMS\_CSS\_LOS\_CLOCKS, [APP\\_MS-S\\_CLOCKS](#), [APP\\_UPA\\_CLOCKS](#), 0, 0)
- [BSP\\_SET\\_L2C\\_CONFIG](#) (0, L2C\_REPL\_LRU, 0, L2C\_MODE\_COPY\_BACK, 0, NULL)

#### Detailed Description

RTEMS configuration Leon header.

#### Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: [common/license.txt](#)

#### Function Documentation

```
BSP_SET_CLOCK ( 12000 , 500000 , 1 , 1 , DEFAULT RTEMS_CSS_LOS_CLOCKS ,  
APP_MSS_CLOCKS , APP_UPA_CLOCKS , 0 , 0 )
```

```
BSP_SET_L2C_CONFIG ( 0 , L2C_REPL_LRU , 0 , L2C_MODE_COPY_BACK , 0 , NULL )
```

#### 0.5.250 `samplers.h` File Reference

```
#include <mvcv_types.h>
```

#### Functions

- void [icvGetRectSubPix\\_8u32f\\_C1R\\_tuned](#) (const u8 \*src, int src\_step, ClSizeW src\_size, u8 \*dst, int dst\_step, ClSizeW win\_size, mvCvPoint2D32fW rect\_center, u8 already\_padded)

#### Function Documentation

```
void icvGetRectSubPix_8u32f_C1R_tuned ( const u8 * src, int src_step, ClSizeW src_size, u8 * dst,  
int dst_step, ClSizeW win_size, mvCvPoint2D32fW rect_center, u8 already_padded )
```

Extracts a rectangle out of the image, using bilinear filtering

## Parameters

in	<i>src</i>	Source image buffer
in	<i>src_step</i>	Source image step
in	<i>src_size</i>	Source image size
out	<i>dst</i>	Destination buffer for the extracted patch
in	<i>dst_step</i>	Destination patch size
in	<i>win_size</i>	Windows size which defines the size of the extracted patch as: win_size.x * 2 x win_size.y * 2 centered in "rect_center"
in	<i>rect_center</i>	Center of the extracted patch
in	<i>inPlace</i>	Specifies if the source image is already in CMX and doesn't need to be DMA-ed in

### 0.5.251 scaleFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.252 scharr\_fp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.253 Sections.h File Reference

#### Macros

- #define **UNIT\_TEST** \_\_attribute\_\_((section(".text.toKeep")))
- #define **CMX\_TEXT** \_\_attribute\_\_((section(".cmx.text")))
- #define **CMX\_DATA** \_\_attribute\_\_((section(".cmx.data")))
- #define **CMX\_CONST\_DATA** \_\_attribute\_\_((section(".cmx.const.data")))
- #define **DDR\_DATA** \_\_attribute\_\_((section(".ddr.data")))
- #define **DDR\_HEAP** \_\_attribute\_\_((section(".ddr.heap")))
- #define **DDR\_BSS** \_\_attribute\_\_((section(".ddr.bss")))
- #define **DDR\_DIRECT\_BSS** \_\_attribute\_\_((section(".ddr\_direct.bss")))
- #define **CMX\_BSS** \_\_attribute\_\_((section(".cmx.bss")))
- #define **CMX\_DATA\_S0** \_\_attribute\_\_((section(".shv0.S.data")))
- #define **CMX\_DATA\_S1** \_\_attribute\_\_((section(".shv1.S.data")))
- #define **CMX\_DATA\_CASCADE\_HID** \_\_attribute\_\_((section(".cmx.data.cascade\_hid")))
- #define **CMX\_DATA\_GLOBAL** \_\_attribute\_\_((section(".cmx.data.global")))
- #define **CMX\_DATA\_INTEGRAL\_IMG** \_\_attribute\_\_((section(".cmx.data.integralImage")))
- #define **LCD\_OVERLAY** \_\_attribute\_\_((section(".ddr\_direct.bss.overlay")))
- #define **OSD** \_\_attribute\_\_((section(".osd.gl2.data")))

## Detailed Description

### Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

## Macro Definition Documentation

```
#define CMX_BSS __attribute__((section(".cmx.bss")))

#define CMX_CONST_DATA __attribute__((section(".cmx.const.data")))

#define CMX_DATA __attribute__((section(".cmx.data")))

#define CMX_DATA_CASCADE_HID __attribute__((section(".cmx.data.cascade_hid")))

#define CMX_DATA_GLOBAL __attribute__((section(".cmx.data.global")))

#define CMX_DATA_INTEGRAL_IMG __attribute__((section(".cmx.data.integralImage")))

#define CMX_DATA_S0 __attribute__((section(".shv0.S.data")))

#define CMX_DATA_S1 __attribute__((section(".shv1.S.data")))

#define CMX_TEXT __attribute__((section(".cmx.text")))

#define DDR_BSS __attribute__((section(".ddr.bss")))

#define DDR_DATA __attribute__((section(".ddr.data")))

#define DDR_DIRECT_BSS __attribute__((section(".ddr_direct.bss")))

#define DDR_HEAP __attribute__((section(".ddr.heap")))

#define LCD_OVERLAY __attribute__((section(".ddr_direct.bss.overlay")))

#define OSD __attribute__((section(".osd.gl2.data")))

#define UNIT_TEST __attribute__((section(".text.toKeep")))
```

### 0.5.254 Sections.h File Reference

#### Macros

- `#define UNIT_TEST __attribute__((section(".text.unittest")))`
- `#define CMX_TEXT __attribute__((section(".cmx.text")))`
- `#define CMX_DATA __attribute__((section(".cmx.data")))`
- `#define CMX_CONST_DATA __attribute__((section(".cmx.const.data")))`
- `#define DDR_DATA __attribute__((section(".ddr.data")))`
- `#define DDR_HEAP __attribute__((section(".ddr.heap")))`
- `#define DDR_BSS __attribute__((section(".ddr.bss")))`
- `#define DDR_DIRECT_BSS __attribute__((section(".ddr_direct.bss")))`
- `#define CMX_BSS __attribute__((section(".cmx.bss")))`

- #define **CMX\_DATA\_S0** \_\_attribute\_\_((section(".shv0.S.data")))
- #define **CMX\_DATA\_S1** \_\_attribute\_\_((section(".shv1.S.data")))
- #define **CMX\_DATA\_CASCADE\_HID** \_\_attribute\_\_((section(".cmx.data.cascade\_hid")))
- #define **CMX\_DATA\_GLOBAL** \_\_attribute\_\_((section(".cmx.data.global")))
- #define **CMX\_DATA\_INTEGRAL\_IMG** \_\_attribute\_\_((section(".cmx.data.integralImage")))
- #define **LCD\_OVERLAY** \_\_attribute\_\_((section(".ddr\_direct.bss.overlay")))
- #define **OSD** \_\_attribute\_\_((section(".osd.gl2.data")))
- #define **CMX\_DMA\_DESCRIPTOR** \_\_attribute\_\_((section(".cmx.cdmaDescriptors")))

## Detailed Description

### Copyright

All code copyright Movidius Ltd 2012, all rights reserved. For License Warranty see: common/license.txt

### Macro Definition Documentation

```
#define CMX_BSS __attribute__((section(".cmx.bss")))

#define CMX_CONST_DATA __attribute__((section(".cmx.const.data")))

#define CMX_DATA __attribute__((section(".cmx.data")))

#define CMX_DATA_CASCADE_HID __attribute__((section(".cmx.data.cascade_hid")))

#define CMX_DATA_GLOBAL __attribute__((section(".cmx.data.global")))

#define CMX_DATA_INTEGRAL_IMG __attribute__((section(".cmx.data.integralImage")))

#define CMX_DATA_S0 __attribute__((section(".shv0.S.data")))

#define CMX_DATA_S1 __attribute__((section(".shv1.S.data")))

#define CMX_DMA_DESCRIPTOR __attribute__((section(".cmx.cdmaDescriptors")))

#define CMX_TEXT __attribute__((section(".cmx.text")))

#define DDR_BSS __attribute__((section(".ddr.bss")))

#define DDR_DATA __attribute__((section(".ddr.data")))

#define DDR_DIRECT_BSS __attribute__((section(".ddr_direct.bss")))

#define DDR_HEAP __attribute__((section(".ddr.heap")))

#define LCD_OVERLAY __attribute__((section(".ddr_direct.bss.overlay")))

#define OSD __attribute__((section(".osd.gl2.data")))

#define UNIT_TEST __attribute__((section(".text.unittest")))
```

### 0.5.255 sLaplacian3x3.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.256 sLaplacian3x3Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.257 sLaplacian5x5.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.258 sLaplacian5x5Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.259 sLaplacian7x7.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.260 sLaplacian7x7Fp16ToFp16.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.261 smoothed\_intensity.h File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

## Data Structures

- struct [mvcvPatternPoint](#)

#### 0.5.262 [sobel.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.263 [ssdPointLine7x7U8U32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.264 [subPixelFilter.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.265 [sumOfAbsDiff11x11.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.266 [sumOfAbsDiff5x5.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.267 [sumOfSquaredDiff11x11.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.268 [sumOfSquaredDiff5x5.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

#### 0.5.269 [sumOfSquaredDiff7x7U8ToU32.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.270 [svd.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.271 [thresholdBinaryRange.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.272 [thresholdBinaryU8.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.273 [thresholdDisparity.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.274 [thresholdDisparityFp16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.275 [thresholdFilter.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.276 [thresholdFilterS16.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.277 [thresholdKernel.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```



## Enumerations

- enum {  
[Thresh\\_To\\_Zero](#) = 0, [Thresh\\_To\\_Zero\\_Inv](#) = 1, [Thresh\\_To\\_Binary](#) = 2, [Thresh\\_To\\_Binary\\_Inv](#) = 3,  
[Thresh\\_Trunc](#) = 4 }

## Enumeration Type Documentation

### anonymous enum

#### Enumerator

***[Thresh\\_To\\_Zero](#)***

***[Thresh\\_To\\_Zero\\_Inv](#)***

***[Thresh\\_To\\_Binary](#)***

***[Thresh\\_To\\_Binary\\_Inv](#)***

***[Thresh\\_Trunc](#)***

### 0.5.278 [tile\\_samplerRGB888.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.279 [tile\\_sampleru10.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.280 [tile\\_sampleru8.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.281 [warpMeshExpand.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.282 [warpMeshSample10bit.h](#) File Reference

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.283 [warpMeshSample8bit.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.284 [warpMeshSample8bitStride.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.285 [warpMeshSampleFp16bit.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.286 [whiteBalanceBayerGBRG.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

### 0.5.287 [whiteBalanceRGB.h File Reference](#)

```
#include <mv_types.h>
#include <mvcv_macro.h>
```

# Index

APP\_MSS\_CLOCKS  
     app\_config.h, 22  
 APP\_UPA\_CLOCKS  
     app\_config.h, 22  
 absoluteDiff.h, 19  
 accumulateFp16.h, 19  
 accumulateFp16MaxPool2s1.h, 19  
 accumulateFp16MaxPool2s2.h, 20  
 accumulateFp16MaxPool3s1.h, 20  
 accumulateFp16MaxPool3s2.h, 20  
 accumulateFp16MaxPool3s2withReLU.h, 20  
 accumulateSquare.h, 20  
 accumulateWeighted.h, 20  
 addV2Fp16.h, 20  
 aggregateAllPaths64.h, 20  
     DISPARITIES, 21  
 aggregateCostSGBM32\_clamp.h, 21  
 aggregateCostSGBM64.h, 21  
 aggregateCostSGBM64\_clamp.h, 21  
 aggregateFivePaths32.h, 21  
 aggregateFivePaths64.h, 21  
 aggregateThreePaths32.h, 21  
 aggregateThreePaths64.h, 21  
 AllOne  
     padKernel\_u16.h, 63  
     padKernel\_u8.h, 64  
 AllZero  
     padKernel\_u16.h, 63  
     padKernel\_u8.h, 64  
 app\_config.h, 22  
     APP\_MSS\_CLOCKS, 22  
     APP\_UPA\_CLOCKS, 22  
     initClocksAndMemory, 22  
 arithmeticAdd.h, 23  
 arithmeticAddmask.h, 23  
 arithmeticSub.h, 23  
 arithmeticSubFp16ToFp16.h, 23  
 arithmeticSubU16.h, 23  
 arithmeticSubmask.h, 23  
 averageV3.h, 23  
 avg1x5kernel  
     kernelUnitTestSample.h, 53  
 avg3x5kernel\_newint  
     kernelUnitTestSample.h, 53  
 avgPool7x7xk.h, 24  
 BSP\_SET\_CLOCK  
     rtems\_config.h, 65  
 BSP\_SET\_L2C\_CONFIG  
     rtems\_config.h, 65  
 bilateral5x5.h, 24  
 bilinearInterpolation.h, 24  
 bilinearInterpolation\_u16.h, 24  
 bilinearInterpolationAligned\_u16.h, 24  
 bitbuffer\_s, 15  
     buf, 16  
     n, 16  
 bitbuffer\_t  
     huffman\_encode.h, 50  
 bitwiseAnd.h, 24  
 bitwiseAndMask.h, 24  
 bitwiseNot.h, 25  
 bitwiseOr.h, 25  
 bitwiseOrMask.h, 25  
 bitwiseXor.h, 25  
 bitwiseXorMask.h, 25  
 BlackPixel  
     padKernel\_u16.h, 63  
     padKernel\_u8.h, 64  
 blendkernel  
     kernelUnitTestSample.h, 53  
 boxFilter.h, 25  
 boxFilter11x11.h, 25  
 boxFilter13x13.h, 25  
 boxFilter15x15.h, 26  
 boxFilter3x3.h, 26  
 boxFilter5x5.h, 26  
 boxFilter7x7.h, 26  
 boxFilter9x9.h, 26  
 buf  
     bitbuffer\_s, 16  
 Buffers.h, 26  
     DDR\_BUFF, 27  
     inputFrameA, 27  
     inputFrameB, 27  
     pyrAL1, 27  
     pyrAL2, 27

- pyrAL3, [27](#)
- pyrAL4, [27](#)
- pyrBL1, [27](#)
- pyrBL2, [27](#)
- pyrBL3, [27](#)
- pyrBL4, [27](#)
- CMX\_BSS
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_CONST\_DATA
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_DATA
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_DATA\_GLOBAL
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_DATA\_S0
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_DATA\_S1
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- CMX\_TEXT
  - opticalFlowPyrLK/unittest/dummy/shared/-  
Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)
- calcBxBy.h, [28](#)
- calcEpipolarDistance.h, [28](#)
- calcG.h, [28](#)
- calcOpticalFlowPyrLK
  - opticalFlowPyrLK.h, [62](#)
- canny.h, [28](#)
- censusMatching16.h, [28](#)
- censusMatching32.h, [28](#)
- censusMatching64.h, [28](#)
- censusMatching65.h, [28](#)
- censusMatchingPyr.h, [29](#)
- censusMatchingPyrOnePos.h, [29](#)
- censusMatchingPyrOnePosWindow.h, [29](#)
- censusMin16.h, [29](#)
- censusMin3.h, [29](#)
- censusMin32.h, [29](#)
- censusMin64.h, [29](#)
- censusMin65.h, [30](#)
- censusMin7.h, [30](#)
- DISPARITIES, [30](#)
- censusMinConfidence32.h, [30](#)
- MVCV\_FUNC, [30](#)
- censusMinConfidence64.h, [31](#)
- MVCV\_FUNC, [31](#)
- censusMinConfidence64Subpixel.h, [31](#)
- censusMinSubpixel3.h, [32](#)
- censusMinSubpixel3Window.h, [32](#)
- censusTransform11x11.h, [32](#)
- censusTransform11x11u8.h, [32](#)
- censusTransform5x5.h, [32](#)
- censusTransformAverageRef7x7.h, [32](#)
- censusTransformAverageRefMask7x7.h, [32](#)
- channelExtract.h, [32](#)
- computeAD32.h, [33](#)
- computeAD64.h, [33](#)
- computeADPyrOnePos.h, [33](#)
- computeADPyrOnePosWindow.h, [33](#)
- computeCombinedCost3.h, [33](#)
- computeCombinedCost32.h, [33](#)
- computeCombinedCost3Window.h, [33](#)
- computeCombinedCost64.h, [33](#)
- confidence
  - ransac\_params, [19](#)
- conv3x3fp32Scharr.h, [34](#)
- convSeparable11x11.h, [41](#)
- convSeparable11x11Fp16ToFp16.h, [41](#)
- convSeparable3x3.h, [41](#)
- convSeparable3x3Fp16ToFp16.h, [41](#)
- convSeparable5x5.h, [41](#)
- convSeparable5x5Fp16ToFp16.h, [41](#)
- convSeparable7x7.h, [41](#)
- convSeparable7x7Fp16ToFp16.h, [41](#)
- convSeparable9x9.h, [42](#)
- convSeparable9x9Fp16ToFp16.h, [42](#)
- convert\_fp16\_u8.h, [34](#)
- convert\_u8\_fp16.h, [34](#)
- convertFp16ToQ4.h, [34](#)
- convertFrom12BppTo8Bpp.h, [34](#)
- convertYUV400ToYUV422.h, [34](#)
- convolution11x11.h, [34](#)
- convolution11x11slhfff.h, [35](#)

- convolution11x11s1xxhx.h, [35](#)
- convolution11x11s2hhhh.h, [35](#)
- convolution11x11s2xxhx.h, [35](#)
- convolution11x11s3hhhh.h, [35](#)
- convolution11x11s4hhhh.h, [35](#)
- convolution11x11s8hhhh.h, [36](#)
- convolution15x1.h, [36](#)
- convolution1x15.h, [36](#)
- convolution1x5.h, [36](#)
- convolution1x5Fp16ToFp16.h, [36](#)
- convolution1x7.h, [36](#)
- convolution1x7Fp16ToFp16.h, [36](#)
- convolution1x9.h, [36](#)
- convolution3x3.h, [37](#)
- convolution3x3Fp16ToFp16.h, [37](#)
- convolution3x3s2hhhh.h, [37](#)
- convolution3x3s2xhhx.h, [37](#)
- convolution3x3s3hhhh.h, [37](#)
- convolution3x3s4hhhh.h, [37](#)
- convolution3x3s8hhhh.h, [37](#)
- convolution5x1.h, [38](#)
- convolution5x1Fp16ToFp16.h, [38](#)
- convolution5x5.h, [38](#)
- convolution5x5Fp16ToFp16.h, [38](#)
- convolution5x5s2hhhh.h, [38](#)
- convolution5x5s3hhhh.h, [38](#)
- convolution5x5s4hhhh.h, [38](#)
- convolution5x5s8hhhh.h, [38](#)
- convolution7x1.h, [39](#)
- convolution7x1Fp16ToFp16.h, [39](#)
- convolution7x7.h, [39](#)
- convolution7x7Fp16ToFp16.h, [39](#)
- convolution7x7Fp16ToU8.h, [39](#)
- convolution7x7s2hhhh.h, [39](#)
- convolution7x7s3hhhh.h, [39](#)
- convolution7x7s4hhhh.h, [39](#)
- convolution7x7s8hhhh.h, [40](#)
- convolution9x1.h, [40](#)
- convolution9x9.h, [40](#)
- convolution9x9Fp16ToFp16.h, [40](#)
- convolution9x9s2hhhh.h, [40](#)
- convolution9x9s3hhhh.h, [40](#)
- convolution9x9s4hhhh.h, [40](#)
- convolution9x9s8hhhh.h, [40](#)
- cornerMinEigenVal.h, [42](#)
- cornerMinEigenVal\_fp32.h, [42](#)
- cornerMinEigenVal\_patched.h, [42](#)
- cvtColorKernelChromaYUV420ToNV12.h, [42](#)
- cvtColorKernelChromaYUV444ToNV12.h, [42](#)
  - inV, [43](#)
  - outUV, [43](#)

- width, [43](#)
- cvtColorKernelRGBToYUV.h, [43](#)
- cvtColorKernelRGBToYUV422.h, [43](#)
- cvtColorKernelYUV422ToRGB.h, [43](#)
- cvtColorKernelYUVToRGB.h, [43](#)
- cvtColorNV21toRGB.h, [43](#)
- cvtColorRGBfp16ToLumaU8.h, [43](#)
- cvtColorRGBfp16ToUV420U8.h, [43](#)
- cvtColorRGBtoChromaNV12.h, [44](#)
- cvtColorRGBtoLuma.h, [44](#)
- cvtColorRGBtoLumaNV12.h, [44](#)
- cvtColorRGBtoNV12.h, [44](#)
- cvtColorRGBtoNV21.h, [44](#)
- cvtColorRGBtoUV.h, [44](#)
- cvtColorRGBtoUV420.h, [44](#)
- DDR\_BSS
  - opticalFlowPyrLK/unittest/dummy/shared/-Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h, [68](#)
- DDR\_BUFF
  - Buffers.h, [27](#)
- DDR\_DATA
  - opticalFlowPyrLK/unittest/dummy/shared/-Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h, [68](#)
- DDR\_DIRECT\_BSS
  - opticalFlowPyrLK/unittest/dummy/shared/-Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h, [68](#)
- DDR\_HEAP
  - opticalFlowPyrLK/unittest/dummy/shared/-Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h, [68](#)
- DISPARITIES
  - aggregateAllPaths64.h, [21](#)
  - censusMin7.h, [30](#)
- data.h, [44](#)
- dc
  - huffman\_s, [17](#)
- dct16bit.h, [44](#)
- Debug
  - RANSAC\_Outputs, [18](#)
- Defines.h, [45](#)
  - HEAP\_SIZE, [45](#)
  - MAX\_FEATURES, [45](#)
  - MAX\_FRAME\_HEIGHT, [45](#)
  - MAX\_FRAME\_WIDTH, [45](#)

- MAX\_PYRAMID\_LEVELS, 45
- SEARCH\_WIN\_SZ, 45
- STOP\_CRIT\_EPS, 45
- STOP\_CRIT\_MAX\_ITER, 45
- dilate.h, 45
- dilate3x3.h, 45
- dilate5x5.h, 45
- dilate7x7.h, 46
- disp2depth.h, 46
- dist\_threshold
  - ransac\_params, 19
- dotProductStreaming
  - f32UnitTestSample.h, 47
- downsampleBilinearLine.h, 46
- dummy\_kernel\_result
  - kernelUnitTestSample.h, 54
- eigenValVec4x4Array.h, 46
- equalizeHist.h, 46
- erode3x3.h, 46
- erode5x5.h, 46
- erode7x7.h, 46
- f32UnitTestSample.h, 47
  - dotProductStreaming, 47
  - mvuDotProductStreaming, 47
- fast9M2.h, 47
- fast9ScoreCv.h, 47
- fast9u16score.h, 47
- fastAtan2Positive.h, 47
- findFundamentalMat
  - FundamentalMatrix.h, 49
- flipKernel.h, 48
- fm
  - RANSAC\_Outputs, 18
- fm8Point
  - fm8Point.h, 48
- fm8Point.h, 48
  - fm8Point, 48
  - halfm8Point, 48
- fundamental\_debug, 16
  - RANSAC\_inliers\_ratio, 16
  - RANSAC\_iterations, 16
  - RANSAC\_nInliers, 16
  - RANSAC\_support, 16
- FundamentalMatrix.h, 48
  - findFundamentalMat, 49
  - POINTS\_PER\_SAMPLE, 49
- gauss.h, 49
- gauss1x5\_u16in\_u32out.h, 49
- gauss5x1\_u32in\_u16out.h, 49
- gaussHx2.h, 49
- gaussHx2\_fp16.h, 49
- gaussVx2.h, 49
- gaussVx2\_fp16.h, 50
- HEAP\_SIZE
  - Defines.h, 45
- hacbit
  - huffman\_s, 17
- haclen
  - huffman\_s, 17
- halfm8Point
  - fm8Point.h, 48
- hammingDistance.h, 50
- harrisResponse.h, 50
- hdcbit
  - huffman\_s, 17
- hdclen
  - huffman\_s, 17
- histogram.h, 50
- huffman\_encode.h, 50
  - bitbuffer\_t, 50
  - huffman\_t, 50
- huffman\_s, 16
  - dc, 17
  - hacbit, 17
  - haclen, 17
  - hdcbit, 17
  - hdclen, 17
  - qtable, 17
- huffman\_t
  - huffman\_encode.h, 50
- icvGetRectSubPix\_8u32f\_C1R\_tuned
  - samplers.h, 65
- inV
  - cvtColorKernelChromaYUV444ToNV12.h, 43
- initClocksAndMemory
  - app\_config.h, 22
- inliers
  - RANSAC\_Outputs, 18
- inliers\_ratio
  - ransac\_params, 19
- inputFrameA
  - Buffers.h, 27
- inputFrameB
  - Buffers.h, 27
- integralImageSquareSumFloatM2.h, 51
- integralImageSquareSumM2.h, 51
- integralImageSumFloatM2.h, 51
- integralImageSumM2.h, 51

- integralImageSumU16U32.h, [51](#)
- interpolatePixelBilinear.h, [51](#)
  - MVCV\_FUNC, [51](#)
- interpolatePixelBilinearS16.h, [52](#)
  - MVCV\_FUNC, [52](#)
- jpegGetBlockY420.h, [52](#)
- kernelUnitTestSample.h, [53](#)
  - avg1x5kernel, [53](#)
  - avg3x5kernel\_newint, [53](#)
  - blendkernel, [53](#)
  - dummy\_kernel\_result, [54](#)
- kernels.h, [52](#)
- LCD\_OVERLAY
  - opticalFlowPyrLK/unittest/dummy/shared/-Sections.h, [67](#)
  - RANSAC/unittest/dummy/shared/Sections.h, [68](#)
- Left
  - padKernel\_u16.h, [63](#)
  - padKernel\_u8.h, [63](#)
- LeftAndRight
  - padKernel\_u16.h, [63](#)
  - padKernel\_u8.h, [63](#)
- localMaxMin3x3\_s16.h, [54](#)
  - MVCV\_FUNC, [54](#)
- lookupTable10to16.h, [55](#)
- lookupTable10to8.h, [55](#)
- lookupTable12to16.h, [55](#)
- lookupTable12to8.h, [55](#)
- lookupTable8to10.h, [55](#)
- lookupTable8to8.h, [55](#)
- MAX\_FEATURES
  - Defines.h, [45](#)
- MAX\_FRAME\_HEIGHT
  - Defines.h, [45](#)
- MAX\_FRAME\_WIDTH
  - Defines.h, [45](#)
- MAX\_NUMPOINTS
  - RANSAC.h, [64](#)
- MAX\_PYRAMID\_LEVELS
  - Defines.h, [45](#)
- MDKdox-MvCv-intro.txt, [58](#)
- MVCV\_FUNC
  - censusMinConfidence32.h, [30](#)
  - censusMinConfidence64.h, [31](#)
  - interpolatePixelBilinear.h, [51](#)
  - interpolatePixelBilinearS16.h, [52](#)
  - localMaxMin3x3\_s16.h, [54](#)
  - maxTest3x3\_fp16.h, [57](#)
  - maxTest3x3\_s16.h, [58](#)
  - meshExpand.h, [59](#)
  - minTest3x3\_fp16.h, [60](#)
  - minTest3x3\_s16.h, [61](#)
  - matmulBT\_xxii.h, [56](#)
  - matrixInv3x3\_fp32.h, [56](#)
  - matrixInv4x4\_fp32.h, [56](#)
  - matrixVectorMultfp16x4.h, [56](#)
  - max\_iterations
    - ransac\_params, [19](#)
  - maxPool2x2s2hh.h, [57](#)
  - maxPool3x3hh.h, [57](#)
  - maxPool3x3s2hh.h, [57](#)
  - maxTest3x3\_fp16.h, [57](#)
    - MVCV\_FUNC, [57](#)
  - maxTest3x3\_s16.h, [58](#)
    - MVCV\_FUNC, [58](#)
  - maximumV2.h, [56](#)
  - maximumV3.h, [56](#)
  - maximumV9.h, [56](#)
  - maximumV9x4.h, [56](#)
  - meanStdDev.h, [58](#)
  - meshExpand.h, [58](#)
    - MVCV\_FUNC, [59](#)
  - meshGenerate.h, [59](#)
  - minAggregate\_line.h, [59](#)
  - minKernel.h, [59](#)
  - minKernel32.h, [59](#)
  - minKernel64.h, [59](#)
  - minMaxKernel.h, [59](#)
  - minMaxPos.h, [60](#)
  - minTest3x3\_fp16.h, [60](#)
    - MVCV\_FUNC, [60](#)
  - minTest3x3\_s16.h, [60](#)
    - MVCV\_FUNC, [61](#)
  - Mirror
    - padKernel\_u16.h, [63](#)
    - padKernel\_u8.h, [64](#)
  - monoImbalance.h, [61](#)
  - mvcvPatternPoint, [17](#)
    - sigma, [17](#)
    - x, [17](#)
    - y, [18](#)
  - mvuDotProductStreaming
    - f32UnitTestSample.h, [47](#)
- n
  - bitbuffer\_s, [16](#)
- nPoints
  - ransac\_params, [19](#)
- nonMax3x3\_fp32.h, [61](#)
- nonMax3x3\_u8.h, [61](#)

OSD  
 opticalFlowPyrLK/unittest/dummy/shared/-  
 Sections.h, [67](#)  
 RANSAC/unittest/dummy/shared/Sections.h,  
[68](#)  
 opticalFlowPyrLK.h, [61](#)  
 calcOpticalFlowPyrLK, [62](#)  
 opticalFlowPyrLK/unittest/dummy/shared/Sections.-  
 h  
 CMX\_BSS, [67](#)  
 CMX\_DATA, [67](#)  
 CMX\_DATA\_S0, [67](#)  
 CMX\_DATA\_S1, [67](#)  
 CMX\_TEXT, [67](#)  
 DDR\_BSS, [67](#)  
 DDR\_DATA, [67](#)  
 DDR\_HEAP, [67](#)  
 LCD\_OVERLAY, [67](#)  
 OSD, [67](#)  
 UNIT\_TEST, [67](#)  
 outUV  
 cvtColorKernelChromaYUV444ToNV12.h,  
[43](#)  
 p1  
 RANSAC\_Inputs, [18](#)  
 p2  
 RANSAC\_Inputs, [18](#)  
 POINTS\_PER\_SAMPLE  
 FundamentalMatrix.h, [49](#)  
 padKernel\_u16.h  
 AllOne, [63](#)  
 AllZero, [63](#)  
 BlackPixel, [63](#)  
 Left, [63](#)  
 LeftAndRight, [63](#)  
 Mirror, [63](#)  
 PixelValue, [63](#)  
 Right, [63](#)  
 WhitePixel, [63](#)  
 padKernel\_u8.h  
 AllOne, [64](#)  
 AllZero, [64](#)  
 BlackPixel, [64](#)  
 Left, [63](#)  
 LeftAndRight, [63](#)  
 Mirror, [64](#)  
 PixelValue, [64](#)  
 Right, [63](#)  
 WhitePixel, [64](#)  
 padKernel\_u16.h, [62](#)  
 padKernel\_u8.h, [63](#)  
 Params  
 RANSAC\_Inputs, [18](#)  
 PixelValue  
 padKernel\_u16.h, [63](#)  
 padKernel\_u8.h, [64](#)  
 pixelPos.h, [64](#)  
 pyrAL1  
 Buffers.h, [27](#)  
 pyrAL2  
 Buffers.h, [27](#)  
 pyrAL3  
 Buffers.h, [27](#)  
 pyrAL4  
 Buffers.h, [27](#)  
 pyrBL1  
 Buffers.h, [27](#)  
 pyrBL2  
 Buffers.h, [27](#)  
 pyrBL3  
 Buffers.h, [27](#)  
 pyrBL4  
 Buffers.h, [27](#)  
 pyrdown.h, [64](#)  
 qtable  
 huffman\_s, [17](#)  
 RANSAC.h, [64](#)  
 MAX\_NUMPOINTS, [64](#)  
 RANSAC\_input, [64](#)  
 RANSAC\_output, [65](#)  
 RANSAC/unittest/dummy/shared/Sections.h  
 CMX\_BSS, [68](#)  
 CMX\_DATA, [68](#)  
 CMX\_DATA\_S0, [68](#)  
 CMX\_DATA\_S1, [68](#)  
 CMX\_TEXT, [68](#)  
 DDR\_BSS, [68](#)  
 DDR\_DATA, [68](#)  
 DDR\_HEAP, [68](#)  
 LCD\_OVERLAY, [68](#)  
 OSD, [68](#)  
 UNIT\_TEST, [68](#)  
 RANSAC\_Inputs, [18](#)  
 p1, [18](#)  
 p2, [18](#)  
 Params, [18](#)  
 RANSAC\_Outputs, [18](#)  
 Debug, [18](#)  
 fm, [18](#)  
 inliers, [18](#)  
 RANSAC\_inliers\_ratio



- fundamental\_debug, 16
- RANSAC\_input
  - RANSAC.h, 64
- RANSAC\_itterations
  - fundamental\_debug, 16
- RANSAC\_nInliers
  - fundamental\_debug, 16
- RANSAC\_output
  - RANSAC.h, 65
- RANSAC\_support
  - fundamental\_debug, 16
- ransac\_params, 19
  - confidence, 19
  - dist\_threshold, 19
  - inliers\_ratio, 19
  - max\_iterations, 19
  - nPoints, 19
- Right
  - padKernel\_u16.h, 63
  - padKernel\_u8.h, 63
- rtems\_config.h, 65
  - BSP\_SET\_CLOCK, 65
- SEARCH\_WIN\_SZ
  - Defines.h, 45
- sLaplacian3x3.h, 69
- sLaplacian3x3Fp16ToFp16.h, 69
- sLaplacian5x5.h, 69
- sLaplacian5x5Fp16ToFp16.h, 69
- sLaplacian7x7.h, 69
- sLaplacian7x7Fp16ToFp16.h, 69
- STOP\_CRIT\_EPS
  - Defines.h, 45
- STOP\_CRIT\_MAX\_ITER
  - Defines.h, 45
- samplers.h, 65
  - icvGetRectSubPix\_8u32f\_C1R\_tuned, 65
- scaleFp16.h, 66
- scharr\_fp16.h, 66
- Sections.h, 66, 67
- sigma
  - mvcvPatternPoint, 17
- smoothed\_intensity.h, 69
- sobel.h, 70
- ssdPointLine7x7U8U32.h, 70
- subPixelFilter.h, 70
- sumOfAbsDiff11x11.h, 70
- sumOfAbsDiff5x5.h, 70
- sumOfSquaredDiff11x11.h, 70
- sumOfSquaredDiff5x5.h, 70
- sumOfSquaredDiff7x7U8ToU32.h, 70
- svd.h, 71
- Thresh\_To\_Binary
  - thresholdKernel.h, 72
- Thresh\_To\_Binary\_Inv
  - thresholdKernel.h, 72
- Thresh\_To\_Zero
  - thresholdKernel.h, 72
- Thresh\_To\_Zero\_Inv
  - thresholdKernel.h, 72
- Thresh\_Trunc
  - thresholdKernel.h, 72
- thresholdKernel.h
  - Thresh\_To\_Binary, 72
  - Thresh\_To\_Binary\_Inv, 72
  - Thresh\_To\_Zero, 72
  - Thresh\_To\_Zero\_Inv, 72
  - Thresh\_Trunc, 72
- thresholdBinaryRange.h, 71
- thresholdBinaryU8.h, 71
- thresholdDisparity.h, 71
- thresholdDisparityFp16.h, 71
- thresholdFilter.h, 71
- thresholdFilterS16.h, 71
- thresholdKernel.h, 71
- tile\_samplerRGB888.h, 72
- tile\_sampleru10.h, 72
- tile\_sampleru8.h, 72
- UNIT\_TEST
  - opticalFlowPyrLK/unittest/dummy/shared/-
    - Sections.h, 67
  - RANSAC/unittest/dummy/shared/Sections.h, 68
- warpMeshExpand.h, 72
- warpMeshSample10bit.h, 72
- warpMeshSample8bit.h, 73
- warpMeshSample8bitStride.h, 73
- warpMeshSampleFp16bit.h, 73
- WhitePixel
  - padKernel\_u16.h, 63
  - padKernel\_u8.h, 64
- whiteBalanceBayerGBRG.h, 73
- whiteBalanceRGB.h, 73
- width
  - cvtColorKernelChromaYUV444ToNV12.h, 43
- x
  - mvcvPatternPoint, 17
- y
  - mvcvPatternPoint, 18