ACF/APEX Kernel SDK

Generated by Doxygen 1.8.8

Fri Dec 14 2018 18:20:12

Contents

1	Main	Page		1
2	Mod	ule Inde	(3
	2.1	Module	·	3
3	Clas	s Index		5
	3.1	Class I	st	5
4	File	Index		7
	4.1	File Lis		7
5	Mod	ule Doo	imentation 1	9
	5.1	UserAl	1	9
		5.1.1	Detailed Description	9
	5.2	Apexc	_pro	0
	5.3	Fast .		1
		5.3.1	Detailed Description	1
		5.3.2	Function Documentation	1
			5.3.2.1 nms3x3	1
	5.4	Image_	proc	3
		5.4.1	Detailed Description	3
	5.5	Histog	m_equalization	4
		5.5.1	Detailed Description	4
		5.5.2	Function Documentation	4
			5.5.2.1 apu_generate_lut	4
			5.5.2.2 apu_histogram_equalization	5
	5.6	Arithm	tic	6
		5.6.1	Detailed Description	6
	5.7	Additio		7
		5.7.1	Detailed Description	7
		5.7.2	Function Documentation	8
			5.7.2.1 add	8
			5.7.2.2 add_in16s_out32s	9

iv CONTENTS

		5.7.2.3	add_in32Q3_28_out32Q3_28	29
		5.7.2.4	add_in32s_out32s	29
		5.7.2.5	add_in32u_out32u	30
		5.7.2.6	add_in64s_out64s	30
		5.7.2.7	add_in64u_out64u	30
		5.7.2.8	apu_add	31
		5.7.2.9	apu_add_in16s_out32s	31
		5.7.2.10	apu_add_in32Q3_28_out32Q3_28	32
		5.7.2.11	apu_add_in32s_out32s	32
		5.7.2.12	apu_add_in64s_out64s	33
		5.7.2.13	apu_disparity	33
		5.7.2.14	apu_gauss_3x1	34
		5.7.2.15	disparity	35
		5.7.2.16	gauss_3x1	35
5.8	Differen	nce		36
	5.8.1	Detailed	Description	36
	5.8.2	Function	Documentation	37
		5.8.2.1	apu_diff_in08u_out16s	37
		5.8.2.2	apu_diff_in16s_out16s	38
		5.8.2.3	apu_diff_in16s_out32s	38
		5.8.2.4	apu_diff_in32s_out32s	38
		5.8.2.5	apu_diff_in64s_out64s	39
		5.8.2.6	difference_filter_in08u_out16s	39
		5.8.2.7	difference_filter_in16s_out16s	40
		5.8.2.8	difference_filter_in16s_out32s	40
		5.8.2.9	difference_filter_in32s_out32s	40
		5.8.2.10	difference_filter_in32s_out64s	41
		5.8.2.11	difference_filter_in32u_out32s	41
		5.8.2.12	difference_filter_in64s_out64s	41
		5.8.2.13	difference_filter_in64u_out64s	43
5.9	Divisio	1		44
	5.9.1	Detailed	Description	44
	5.9.2	Function	Documentation	45
		5.9.2.1	apu_dot_division	45
		5.9.2.2	apu_dot_division_N64s_D32s_Q64s	45
		5.9.2.3	apu_dot_inv_NewtonRaphson	46
		5.9.2.4	apu_dot_log2	47
		5.9.2.5	compute64bitLog2	48
		5.9.2.6	compute64bitLog2u	48
		5.9.2.7	computeInv_NewtonRaphson	48

CONTENTS

		5.9.2.8	computeLog2	49
		5.9.2.9	computeLog2u	49
		5.9.2.10	dot_division_filter	49
		5.9.2.11	dot_division_filter_N64s_D32s_Q64s	50
5.10	Multipli	cation		51
	5.10.1	Detailed I	Description	51
	5.10.2	Function	Documentation	54
		5.10.2.1	apu_dot_left_shift_in16u_out16s	54
		5.10.2.2	apu_dot_lsh1_in32s_out32s	54
		5.10.2.3	apu_dot_lsh1_in32s_out64s	55
		5.10.2.4	apu_dot_lsh1_in32s_Q3_28_out64s	55
		5.10.2.5	apu_dot_mult_in16s_out32s	56
		5.10.2.6	apu_dot_mult_in32s_in16s_out32s	56
		5.10.2.7	apu_dot_mult_in32s_out32s	57
		5.10.2.8	apu_dot_mult_in32s_out64s	57
		5.10.2.9	apu_dot_mult_scalar_in08u_out16s	58
		5.10.2.10	apu_dot_mult_scalar_in32s_out32s	58
		5.10.2.11	apu_dot_right_shift_in64s_out32s	59
		5.10.2.12	apu_dot_right_shift_in64s_out64s	59
		5.10.2.13	change64bitSign	60
		5.10.2.14	dot_mult_in16s_out32s_filter	60
		5.10.2.15	dot_mult_in32s_in16s_out32s_filter	60
		5.10.2.16	dot_mult_in32s_out32s_filter	61
		5.10.2.17	dot_mult_in32s_out64s_filter	61
		5.10.2.18	dot_mult_in32u_out64u_filter	62
		5.10.2.19	dot_mult_in64s_out64s_filter	62
		5.10.2.20	dot_mult_in64u_out64u_filter	62
		5.10.2.21	dot_mult_scalar_in08u_out16s_filter	63
		5.10.2.22	dot_mult_scalar_in32s_out32s_filter	63
		5.10.2.23	hasSign	64
		5.10.2.24	lsh_in16u_out16s_filter	64
		5.10.2.25	lsh_in32s_out32s_filter	64
		5.10.2.26	lsh_in32s_out64s_filter	65
		5.10.2.27	lsh_in32s_Q3_28_out64s_filter	65
		5.10.2.28	lsh_in32u_out32u_filter	65
		5.10.2.29	lsh_in32u_out64u_filter	66
		5.10.2.30	rsh_in32s_out32s_filter	66
		5.10.2.31	rsh_in32u_out32u_filter	66
		5.10.2.32	rsh_in64s_out32s_filter	67
		5.10.2.33	rsh_in64s_out64s_filter	67

vi CONTENTS

	5.10.2.34 rsh_in64u_out64u_filter
5.11 Squ	are
5.11	.1 Detailed Description
5.11	.2 Function Documentation
	5.11.2.1 apu_dot_sqr_in16s_out32u
	5.11.2.2 apu_dot_sqr_in32s_out32u
	5.11.2.3 apu_dot_sqr_in32s_out64u
	5.11.2.4 dot_sqr_in16s_out32u_filter
	5.11.2.5 dot_sqr_in32s_out32u_filter
	5.11.2.6 dot_sqr_in32s_out64u_filter
	5.11.2.7 dot_sqr_in32u_out64u_filter
	5.11.2.8 dot_sqr_in64s_out64u_filter
	5.11.2.9 dot_sqr_in64u_out64u_filter
	5.11.2.10 vsqrt_32
5.12 Max	imum
5.12	.1 Detailed Description
5.12	.2 Function Documentation
	5.12.2.1 apu_max
	5.12.2.2 max
5.13 Com	parison
5.13	.1 Detailed Description
5.14 Com	parison
5.14	.1 Detailed Description
5.14	.2 Function Documentation
	5.14.2.1 absLower_in32s
	5.14.2.2 absLower_in32s_scalar16u
	5.14.2.3 and_3Pt_in16u_out16u
	5.14.2.4 and_in08u_in16u_out16u
	5.14.2.5 and_in08u_out16u
	5.14.2.6 and_in16u_out16u
	5.14.2.7 and_kn
	5.14.2.8 apu_abs_lower_in32s
	5.14.2.9 apu_abs_lower_in32s_scalar16u
	5.14.2.10 apu_and
	5.14.2.11 apu_and_3Pt_in16u_out16u
	5.14.2.12 apu_and_in08u_in16u_out16u
	5.14.2.13 apu_and_in08u_out16u
	5.14.2.14 apu_and_in16u_out16u
	5.14.2.15 apu_lower
	5.14.2.16 apu_lower_in16s

CONTENTS vii

		5.14.2.17 apu_lower_in32s	38
		5.14.2.18 apu_lower_in64s	39
		5.14.2.19 apu_lower_in64u	39
		5.14.2.20 apu_lower_scalar	90
		5.14.2.21 apu_lowerEqual_in32s	90
		5.14.2.22 apu_mask8b	91
		5.14.2.23 lower	91
		5.14.2.24 lower_in16s	91
		5.14.2.25 lower_in32s	92
		5.14.2.26 lower_in64s	92
		5.14.2.27 lower_in64u	92
		5.14.2.28 lower_scalar	93
		5.14.2.29 lowerEqual_in32s	93
		5.14.2.30 mask_kn	93
5.15	Binary	descriptor matching	95
	5.15.1	Detailed Description	95
	5.15.2	Function Documentation	95
		5.15.2.1 apu_match_descriptors	95
		5.15.2.2 Match	96
5.16	Conver	sion S	97
	5.16.1	Detailed Description	97
5.17	Bit widt	th conversion	98
	5.17.1	Detailed Description	98
	5.17.2	Function Documentation	98
		5.17.2.1 apu_16low_to_8	98
		5.17.2.2 f16low_to_8	98
5.18	Color c	onversion)0
	5.18.1	Detailed Description)0
	5.18.2	Function Documentation)1
		5.18.2.1 apu_rgb_to_grayscale)1
		5.18.2.2 apu_rgb_to_hsv_hue_sat)1
		5.18.2.3 apu_rgb_to_hsv_hue_sat_grey)2
		5.18.2.4 apu_rgb_to_hsv_sat)3
		5.18.2.5 apu_rgb_to_hsv_svr)3
		5.18.2.6 rgb_to_grayscale)4
		5.18.2.7 rgb_to_hsv_hue_sat)4
		5.18.2.8 rgb_to_hsv_hue_sat_grey)4
		5.18.2.9 rgb_to_hsv_sat)5
		5.18.2.10 rgb_to_hsv_svr)5
5.19	Display	/)6

viii CONTENTS

	5.19.1	Detailed Description
	5.19.2	Function Documentation
		5.19.2.1 apu_mark_color_channel
5.20	Marking	g on images
	5.20.1	Detailed Description
	5.20.2	Function Documentation
		5.20.2.1 apu_mark
		5.20.2.2 mark
		5.20.2.3 mark_color_channel
5.21	Feature	e detection
	5.21.1	Detailed Description
5.22	Fast9 fe	eature detection
	5.22.1	Detailed Description
	5.22.2	Function Documentation
		5.22.2.1 apu_fast9
		5.22.2.2 apu_fast9_unsuppressed_score
5.23	Harris	corner detection
	5.23.1	Detailed Description
	5.23.2	Function Documentation
		5.23.2.1 apu_harris
		5.23.2.2 apu_harris
		5.23.2.3 apuHarrisResponse
		5.23.2.4 apuHarrisResponse
5.24	Sum of	Absolute Differences
	5.24.1	Detailed Description
	5.24.2	Function Documentation
		5.24.2.1 apu_sad
		5.24.2.2 apu_sad_impl
5.25	Filtering	g
	5.25.1	Detailed Description
5.26	Col_filt	er
	5.26.1	Detailed Description
	5.26.2	Function Documentation
		5.26.2.1 col_filter
		5.26.2.2 col_filter
	5.26.3	Variable Documentation
		5.26.3.1 COL_PADDING
		5.26.3.2 FILTER_COLS
5.27	Correla	tion
	5.27.1	Detailed Description

CONTENTS

5.27.2	Function Documentation
	5.27.2.1 apu_correlation
	5.27.2.2 apu_correlation_1x3
	5.27.2.3 apu_correlation_3x1
	5.27.2.4 apu_correlation_3x3
	5.27.2.5 apu_correlation_5x5
	5.27.2.6 apu_correlation_7x7
	5.27.2.7 apu_gradient_x
	5.27.2.8 apu_gradient_y
	5.27.2.9 apu_scharr_x
	5.27.2.10 apu_scharr_y
	5.27.2.11 correlationantisymXfilter
	5.27.2.12 correlationantisymXsymYfilter
	5.27.2.13 correlationantisymXsymYfilter_16s
	5.27.2.14 correlationantisymXYfilter
	5.27.2.15 correlationantisymYfilter
	5.27.2.16 correlationsymXantisymYfilter
	5.27.2.17 correlationsymXantisymYfilter_16s
	5.27.2.18 correlationsymXfilter
	5.27.2.19 correlationsymXYfilter
	5.27.2.20 correlationsymYfilter
	5.27.2.21 correlation_filter
	5.27.2.22 initFilters
	5.27.2.23 performCorrelation
5.28 filterin	g
5.28.1	Detailed Description
5.28.2	Function Documentation
	5.28.2.1 apu_filter_a
	5.28.2.2 apu_filter_a_impl
5.29 Media	n filtering
5.29.1	Detailed Description
5.29.2	Function Documentation
	5.29.2.1 apu_flt_median_3x3
	5.29.2.2 median_3x3_8bpp
5.30 Sobel	Filtering
5.30.1	Detailed Description
5.30.2	Function Documentation
	5.30.2.1 apu_flt_sobel_3x3
	5.30.2.2 apu_flt_sobel_3x3_x
	5.30.2.3 apu_flt_sobel_3x3_y

X CONTENTS

		5.30.2.4	sobel_3x3_8bpp .		 	 	 	 	 144
5.31	Gaussi	an filtering			 	 	 	 	 145
	5.31.1	Detailed	Description		 	 	 	 	 145
	5.31.2	Function	Documentation		 	 	 	 	 145
		5.31.2.1	apu_gauss_3x3 .		 	 	 	 	 145
		5.31.2.2	apu_gauss_3x3 .		 	 	 	 	 146
		5.31.2.3	apu_gauss_5x5 .		 	 	 	 	 146
		5.31.2.4	Gauss_5x5filter		 	 	 	 	 146
5.32	Image	gradient			 	 	 	 	 147
	5.32.1	Detailed	Description		 	 	 	 	 147
	5.32.2	Function	Documentation		 	 	 	 	 148
		5.32.2.1	apu_gr_abs		 	 	 	 	 148
		5.32.2.2	apu_gradient		 	 	 	 	 148
		5.32.2.3	apu_gradient_abs		 	 	 	 	 148
		5.32.2.4	apuGradAbs		 	 	 	 	 149
		5.32.2.5	apuGradient		 	 	 	 	 149
		5.32.2.6	apuGradient_out08	s	 	 	 	 	 149
		5.32.2.7	apuGradientAbs .		 	 	 	 	 150
5.33	Non-ma	aximum sı	ppression		 	 	 	 	 151
	5.33.1	Detailed	Description		 	 	 	 	 151
	5.33.2	Function	Documentation		 	 	 	 	 151
		5.33.2.1	apu_nms		 	 	 	 	 151
		5.33.2.2	apu_nms16		 	 	 	 	 152
		5.33.2.3	apu_nms16		 	 	 	 	 152
		5.33.2.4	apu_nms_impl		 	 	 	 	 152
5.34	Row_fil	lter			 	 	 	 	 153
	5.34.1	Detailed	Description		 	 	 	 	 153
	5.34.2	Function	Documentation		 	 	 	 	 153
		5.34.2.1	row_filter		 	 	 	 	 153
		5.34.2.2	row_filter_impl		 	 	 	 	 154
5.35	Saturat	ion			 	 	 	 	 155
	5.35.1	Detailed	Description		 	 	 	 	 155
	5.35.2	Function	Documentation		 	 	 	 	 155
		5.35.2.1	apu_binarize		 	 	 	 	 155
		5.35.2.2	apu_mask		 	 	 	 	 156
		5.35.2.3	apu_saturate_nonz	ero	 	 	 	 	 156
		5.35.2.4	binarize		 	 	 	 	 156
		5.35.2.5	mask		 	 	 	 	 157
		5.35.2.6	saturate_nonzero		 	 	 	 	 157
5.36	Geome	etry			 	 	 	 	 158

CONTENTS xi

	5.36.1	Detailed Description	158
5.37	Remap	bilinear	159
	5.37.1	Detailed Description	159
	5.37.2	Function Documentation	159
		5.37.2.1 remap_bilinear_grayscale	159
		5.37.2.2 remap_bilinear_grayscale_impl	160
		5.37.2.3 remap_bilinear_rgb	160
		5.37.2.4 remap_bilinear_rgb_impl	161
5.38	Remap	nearest	162
	5.38.1	Detailed Description	162
	5.38.2	Function Documentation	162
		5.38.2.1 remap_nearest_grayscale	162
		5.38.2.2 remap_nearest_grayscale_impl	163
5.39	Rotate	image by 180 deg	164
	5.39.1	Detailed Description	164
	5.39.2	Function Documentation	164
		5.39.2.1 apu_rotate_180	164
		5.39.2.2 rotate_180	165
5.40	Morpho	ology	166
	5.40.1	Detailed Description	166
5.41	Dilation	1	167
	5.41.1	Detailed Description	167
	5.41.2	Function Documentation	167
		5.41.2.1 apu_dilate_diamond	167
		5.41.2.2 dilate_diamond	167
5.42	Object	detection	169
	5.42.1	Detailed Description	169
5.43	Haar ca	ascade object detection	170
	5.43.1	Detailed Description	170
	5.43.2	Function Documentation	171
		5.43.2.1 apu_haar_cascade	171
		5.43.2.2 haar_cascade	172
5.44	LBP ca	ascade object detection	174
	5.44.1	Detailed Description	174
	5.44.2	Function Documentation	175
		5.44.2.1 apu_lbp_cascade	175
		5.44.2.2 lbp_cascade	176
5.45	Optimiz	zation	178
	5.45.1	Detailed Description	178
5.46	Summe	ed area table	179

xii CONTENTS

	5.46.1	Detailed Description	79
	5.46.2	Function Documentation	79
		5.46.2.1 apu_sat	79
		5.46.2.2 sat32	79
5.47	SAT-ba	sed box filter	81
	5.47.1	Detailed Description	81
	5.47.2	Function Documentation	81
		5.47.2.1 apu_sat_box_filter	81
		5.47.2.2 sat_box_filter_impl	82
5.48	Resizin	g	83
	5.48.1	Detailed Description	83
5.49	Image	downsampling	84
	5.49.1	Detailed Description	84
	5.49.2	Function Documentation	84
		5.49.2.1 apu_downsample	84
		5.49.2.2 apu_downsample_16u	85
		5.49.2.3 apu_downsample_gauss	85
		5.49.2.4 downsample	85
		5.49.2.5 downsample_16u	86
		5.49.2.6 downsample_gauss	86
5.50	Image	upsampling	87
	5.50.1	Detailed Description	87
	5.50.2	Function Documentation	87
		5.50.2.1 apu_upsample	87
		5.50.2.2 upsample	87
5.51	Statistic	cs	89
	5.51.1	Detailed Description	89
	5.51.2	Function Documentation	90
		5.51.2.1 apu_columns_sum	90
5.52		ulation	
	5.52.1	Detailed Description	92
	5.52.2	Function Documentation	92
		5.52.2.1 accumulation_in32s_filter	92
		5.52.2.2 accumulation_in32u_filter	93
		5.52.2.3 apu_accumulation	93
5.53	Column	n_sum	95
	5.53.1	Detailed Description	95
	5.53.2	Function Documentation	95
		5.53.2.1 column_sum	
5.54	Histofg	rad	96

CONTENTS xiii

			D. C. D. C. C.	
			Detailed Description	
		5.54.2	Function Documentation	
			5.54.2.1 apu_histofgrad	
	5.55	_	ents	
			Detailed Description	
		5.55.2	Function Documentation	97
			5.55.2.1 apu_histogram	97
			5.55.2.2 hist	98
			5.55.2.3 hog	98
	5.56	Reduct	on	99
		5.56.1	Detailed Description	99
		5.56.2	Function Documentation	99
			5.56.2.1 apu_reduction	99
			5.56.2.2 apu_reduction_for_clmn	200
			5.56.2.3 reduc	200
			5.56.2.4 reduc	200
6	Clas			203
	6.1	APEX_	HaarCascadeFeature Struct Reference	203
	6.2	APEX_	HaarCascadeStage Struct Reference	203
	6.3	APEX_	bpFeature Struct Reference	203
	6.4	APEX_	bpStage Struct Reference	204
	6.5	point_t	Struct Reference	204
	6.6	rect_t S	Struct Reference	204
	6.7	RESIZI	E_DESCRIPTOR Struct Reference	204
		6.7.1	Detailed Description	205
		6.7.2	Member Data Documentation	205
			6.7.2.1 dst_bw	205
			6.7.2.2 fbnk	205
			6.7.2.3 out_round	205
			6.7.2.4 scl_fact	205
			6.7.2.5 taps	205
			6.7.2.6 unused	205
	6.8	resize_	descriptor Struct Reference	206
7	File I	Docume	ntation 2	207
	7.1		oamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234↔ ernels/apu/apexcv_gdc_ldw2/src/cg_kernel.h File Reference	207
		7.1.1	Detailed Description	207
	7.2		pamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234c- prnels/apu/apexcv_pro_remap/src/cg_kernel.h File Reference	207

XIV

	7.2.1	Detailed I	Description	207
7.3			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_resize/src/cg_kernel.h File Reference	
	7.3.1	Detailed I	Description	207
7.4			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_affine/src/affine_transform_acf.cpp File Reference	
	7.4.1		Description	
	7.4.1		Documentation	
	7.4.2	7.4.2.1	affine bilinear interpolate	
7.5	/homo/		amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	
7.5			amboo-nome-dif/xmi-data/build-dif/AVS-VSDKIN9-3OBT/vSdk/export/Sdk/s52v254-4/apexcv_pro_affine/src/affine_transform_acf.h File Reference	
	7.5.1	Detailed I	Description	210
7.6			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_affine/src/affine_transform_apu.cpp File Reference	
	7.6.1	Detailed I	Description	210
7.7			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_brief/src/brief_acf.cpp File Reference	
	7.7.1	Detailed I	Description	212
	7.7.2	Function	Documentation	212
		7.7.2.1	compute_brief_descriptor	212
7.8	/home/	bamboo/ba	amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-	_
	_sdk/ke		/apexcv_pro_brief/src/brief_acf.h File Reference	
	7.8.1		Description	
7.9			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_canny/src/canny_acf.cpp	
	7.9.1	Detailed I	Description	215
	7.9.2	Function	Documentation	215
		7.9.2.1	canny_create_image	215
		7.9.2.2	canny_nms_promote	215
		7.9.2.3	canny_non_maxima_suppress	216
		7.9.2.4	canny_promote_edges	216
		7.9.2.5	canny_promote_edges_full	216
7.10			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_canny/src/canny_acf.h File Reference	
	7.10.1	Detailed I	Description	217
7.11			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_canny/src/canny_apu.cpp File Reference	
	7.11.1	Detailed I	Description	218
7.12			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_canny/src/canny_apu.h File Reference	
		•	Description	
			Documentation	
		7.12.2.1	apu_canny_connect_edges	219

CONTENTS xv

	7.12.2.2 apu_canny_connect_edges_full	219
	7.12.2.3 apu_canny_create_image	219
	7.12.2.4 apu_canny_suppress	219
7.13	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_fast/src/fast_acf.cpp File Reference$	
7.14	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.cpp File Reference$	
	7.14.1 Detailed Description	224
7.15	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.h \ File \ Reference \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	
	7.15.1 Detailed Description	226
7.16	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_acf.cpp File Reference$	
	7.16.1 Detailed Description	228
7.17	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_acf.h File Reference$	
	7.17.1 Detailed Description	230
7.18	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_hog/src/hog_apu.h$ File Reference	
	7.18.1 Detailed Description	230
7.19	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.cpp File Reference$	
	7.19.1 Detailed Description	232
	7.19.2 Function Documentation	232
	7.19.2.1 horizontal_gaus_laplacian_mid	232
	7.19.2.2 vertical_gaus_laplacian_last	232
	7.19.2.3 vertical_gaus_laplacian_mid	
7.20	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.h File Reference$	
	7.20.1 Detailed Description	234
7.21	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.cpp File Reference$	_ 234
	7.21.1 Detailed Description	234
7.22	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.h File Reference$	_ 234
	7.22.1 Detailed Description	235
	7.22.2 Image Pyramid Creation	235
	7.22.3 Function Documentation	235
	7.22.3.1 apu_pyr_horizontal_gaus_laplacian	235
	7.22.3.2 apu_pyr_vertical_gaus_laplacian_last	235
	7.22.3.3 apu_pyr_vertical_gaus_laplacian_mid	236
7.23	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.cpp_File_Reference	_ 236

xvi CONTENTS

	7.23.1	Detailed Description	238
	7.23.2	Function Documentation	238
		7.23.2.1	238
		7.23.2.2 Ikpyramid_ht_centraldxdy_7x7	239
		7.23.2.3 Ikpyramid_templ_bilinear_08u_7x7	239
7.24		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	
		ernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.h File Reference	
7.05		Detailed Description	
7.25		ernels/apu/apexcv_pro_lktracker/src/lktracker_acf.cpp File Reference	
	7.25.1	Detailed Description	242
	7.25.2	Function Documentation	242
		7.25.2.1 ktracker_core_7x7	242
		7.25.2.2 lktracker_ht_centraldxdy_7x7	243
		7.25.2.3 ktracker_templ_bilinear_08u_7x7	243
7.26		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	
	_	ernels/apu/apexcv_pro_lktracker/src/lktracker_acf.h File Reference	
		Detailed Description	
7.27		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ernels/apu/apexcv_pro_pyramid/src/pyramid_acf.cpp File Reference	
	7.27.1	Detailed Description	246
	7.27.2	Function Documentation	246
		7.27.2.1 horizontal_gaus	246
		7.27.2.2 horizontal_gaus_and_expand	246
		7.27.2.3 vertical_gaus	247
		7.27.2.4 vertical_gaus_and_reduce	247
7.28		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-ernels/apu/apexcv_pro_pyramid/src/pyramid_acf.h File Reference	
		Detailed Description	
7.29	/home/l	bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	د
	_sdk/ke	ernels/apu/apexcv_pro_pyramid/src/pyramid_apu.cpp File Reference	248
	7.29.1	Detailed Description	249
7.30		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-ernels/apu/apexcv_pro_pyramid/src/pyramid_apu.h File Reference	
	7.30.1	Detailed Description	249
7.31		bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ernels/apu/apexcv_pro_remap/src/remap_apu.h File Reference	
		Detailed Description	
		Function Documentation	
		7.31.2.1 remap_bilinear_grayscale	249
		7.31.2.2 remap_bilinear_rgb	250
7.32		. –	د

CONTENTS xvii

	7.32.1	Detailed I	Description	251
	7.32.2	Macro De	efinition Documentation	251
		7.32.2.1	FLT_BANK_SIZE	251
		7.32.2.2	FLT_DOWN_SAMPLING1	251
		7.32.2.3	FLT_DOWN_SAMPLING2	251
		7.32.2.4	FLT_UP_SAMPLING	251
		7.32.2.5	RESIZE_DESC_SIZE	251
		7.32.2.6	RSZ_DECIMAL_SCL	252
7.33			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /apexcv_pro_resize/src/resize_definitions_apu.h File Reference	
	7.33.1	Detailed I	Description	252
	7.33.2	Macro De	efinition Documentation	253
		7.33.2.1	FLT_BANK_SIZE	253
		7.33.2.2	FLT_DOWN_SAMPLING1	253
		7.33.2.3	FLT_DOWN_SAMPLING2	253
		7.33.2.4	FLT_UP_SAMPLING	253
		7.33.2.5	RESIZE_DESC_SIZE	253
		7.33.2.6	RSZ_DECIMAL_SCL	253
7.34			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /dnn/src/apexdnn_sn_acf.cpp File Reference	
	7.34.1	Detailed I	Description	256
	7.34.2	Function	Documentation	256
		7.34.2.1	apu_conv3x3mps1_module_nhcw_forward	256
7.35			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /dnn/src/apexdnn_sn_acf.h File Reference	
	7.35.1	Detailed I	Description	258
7.36			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /dnn/src/neural_network_acf.cpp File Reference	
			Description	
	7.36.2	Function	Documentation	262
		7.36.2.1	apu_fire2_conv1mps1_forward	263
7.37			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- /dnn/src/neural_network_acf.h File Reference	
	7.37.1	Detailed I	Description	265
7.38			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-/sample_arithmetic_kernels/src/add_apu.h File Reference	
	7.38.1	Detailed I	Description	266
7.39			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-/sample_arithmetic_kernels/src/difference_acf.cpp File Reference	
7.40			amboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-/sample_arithmetic_kernels/src/difference_apu.h File Reference	
	7.40.1	Detailed I	Description	269

xviii CONTENTS

7.41	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_acf.cpp File Reference	
7.42	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_apu.h File Reference	
	7.42.1 Detailed Description	271
7.43	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow \\ _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_acf.cpp File Reference$	
7.44	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_apu.h File Reference$	
	7.44.1 Detailed Description	275
7.45	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_sqr_acf.cpp File Reference$	
7.46	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_sqr_apu.h File Reference$	
	7.46.1 Detailed Description	277
7.47	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow \\ _sdk/kernels/apu/sample_arithmetic_kernels/src/max_acf.cpp File Reference$	
7.48	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_arithmetic_kernels/src/max_apu.h File Reference$	
	7.48.1 Detailed Description	279
7.49	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.cpp File Reference$	
	7.49.1 Detailed Description	280
	7.49.2 Function Documentation	280
	7.49.2.1 apu_flt_box_generic_16s	280
	7.49.2.2 harris_box_5x5_16s	281
7.50	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.h \ File \ Reference \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $	
	7.50.1 Detailed Description	282
7.51	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/comparison_acf.cpp File Reference$	
7.52	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.cpp File Reference$	
	7.52.1 Detailed Description	285
7.53	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.h File Reference$	
	7.53.1 Detailed Description	287
7.54	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_acf.cpp File Reference$	
7.55	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.cpp File Reference$	
	7.55.1 Detailed Description	288
7.56	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.h File Reference$	289
	7.56.1 Detailed Description	289

CONTENTS xix

7.07	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234←_sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_acf.cpp File Reference	289
7.58	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.cpp File Reference$	290
	7.58.1 Detailed Description	290
7.59	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.h$ File Reference	
7.60	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_acf.cpp File Reference$	
7.61	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.cpp File Reference	
	7.61.1 Detailed Description	292
7.62	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-c_sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.h File Reference	
7.63	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_acf.cpp File Reference$	
7.64	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.cpp File Reference$	
	7.64.1 Detailed Description	295
7.65	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.h$ File Reference	
7.66	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_display_kernels/src/mark_acf.cpp File Reference$	
7.67	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_display_kernels/src/mark_apu.cpp File Reference$	
	7.67.1 Detailed Description	297
7.68	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_display_kernels/src/mark_apu.h File Reference$	297
	7.68.1 Detailed Description	298
7.69	7.68.1 Detailed Description	.
	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	298
	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	298 299
7.70	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File Reference	298 299 299
7.70	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File Reference 7.70.1 Detailed Description	298 299 299 299
7.70 7.71	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File Reference 7.70.1 Detailed Description	298 299 299 299 300
7.70 7.71 7.72	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File Reference 7.70.1 Detailed Description	298 299 299 299 300
7.70 7.71 7.72	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference	298 299 299 299 300
7.70 7.71 7.72 7.73	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference	298 299 299 299 300 300 301 301

CONTENTS

7.75	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 c_sdk/kernels/apu/sample_feature_detection_kernels/src/sad_acf.cpp File Reference	302
7.76	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_feature_detection_kernels/src/sad_apu.h File Reference	302
	7.76.1 Detailed Description	303
7.77	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_acf.cpp File Reference$	
7.78	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.cpp File Reference$	304
	7.78.1 Detailed Description	304
7.79	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.h File Reference$	305
	7.79.1 Detailed Description	305
	7.79.2 The Column Filter	305
7.80	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/correlation_acf.cpp File Reference	
7.81	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/correlation_apu.cpp File Reference	308
	7.81.1 Detailed Description	309
7.82	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-csdk/kernels/apu/sample_filtering_kernels/src/correlation_apu.h File Reference	309
	7.82.1 Detailed Description	310
7.83	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/filter_a_acf.cpp File Reference	
7.84	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.cpp File Reference	
	7.84.1 Detailed Description	311
7.85	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.h File Reference	
	7.85.1 Detailed Description	312
7.86	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_acf.cpp File Reference	
7.87	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.cpp File Reference	313
	7.87.1 Detailed Description	313
7.88	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.h File Reference	314
	7.88.1 Detailed Description	314
7.89	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_acf.cpp File Reference	
7.90	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.cpp File Reference	
	7.90.1 Detailed Description	315
7.91	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.h File Reference	

CONTENTS xxi

	7.91.1 Detailed Description	316
7.92	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_acf.cpp File Reference$	
7.93	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_apu.h File Reference$	
	7.93.1 Detailed Description	318
7.94	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_acf.cpp File Reference	
7.95	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow \\ _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_apu.cpp File Reference$	
	7.95.1 Detailed Description	319
7.96	$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_apu.h File Reference$	
	7.96.1 Detailed Description	320
7.97	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_acf.cpp File Reference	
7.98	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.cpp File Reference	
	7.98.1 Detailed Description	321
7.99	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-csdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.h File Reference	
	7.99.1 Detailed Description	322
7.100	0/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gradient_acf.cpp File Reference	
7.10	1/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.cpp File Reference	
	7.101.1 Detailed Description	324
7.102	2/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.h File Reference	
	7.102.1 Detailed Description	325
7.103	3/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/nms_acf.cpp File Reference	
7.104	4/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.cpp File Reference	
	7.104.1 Detailed Description	326
7.105	5/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.h File Reference	
	7.105.1 Detailed Description	327
7.106	6/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_acf.cpp File Reference	
7.107	7/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_apu.cpp File Reference	
	7.107.1 Detailed Description	328
7.108	B/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample filtering kernels/src/row filter a apu.h File Reference	328

xxii CONTENTS

7.110/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.cpp File Reference 336 7.111.0 Letailed Description 330 7.111/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.h File Reference 330 7.111.1 Detailed Description 331 7.112/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/disparity_act.cpp File Reference 331 7.113/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/disparity_apu.h File Reference 332 7.113.1 Detailed Description 333 7.114/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_apu.h File Reference 333 7.115/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.cpp File Reference 334 7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.cpp File Reference 335 7.116.1 Detailed Description 336 7.117/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKINS-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_act.cpp File Reference 336 7.118.1 Detailed Description 336 7.119.1 Detailed Description 336 7.119.1 Detailed Description 337 7.119.1 Detailed Description 338 7.119.1 Detailed Description 338 7.111.1 Detailed Description 338 7.111.1 Detailed Description 339 7.111.1 Detailed Description 339 7.111.1 Detailed Description 339 7.112.1 Detailed Description 339 7.112.1 Detailed Description 339 7.113.1 Detailed Description 339 7.114.1 Detailed Description 339 7.115.1 Detailed Description 340 7.122.1 Deta	7.108.1 Detailed Description	329
	·	
7.111/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.h File Reference 33C 7.112/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/disparity_act.cpp File Reference 331 7.113/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/disparity_apu.h File Reference 332 7.113.home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.pp File Reference 333 7.114.home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.pp File Reference 34 7.115./home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference 335 7.116.1 Detailed Description 336 7.117/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/renap_nearest_apu.h File Reference 336 7.116.home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_act.pp File Reference 336 7.118.home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.pp File Reference 337 7.118.1 Detailed Description 337 7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.pp File Reference 336 7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.pp File Reference 346 7.121/home/ba		
	7.110.1 Detailed Description	330
7.112/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/disparity_act.cpp File Reference		
sdk/kernels/apu/sample_geometry_kernels/src/disparity_act.cpp File Reference	7.111.1 Detailed Description	331
sdk/kernels/apu/sample_geometry_kernels/src/disparity_apu.h File Reference	·	
7.114/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_apu.h File Reference 333 7.115/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.cpp File Reference 334 7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference 335 7.116.1 Detailed Description 335 7.117/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_act.cpp File Reference 336 7.118/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp File Reference 337 7.118.1 Detailed Description 337 7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.b File Reference 336 7.119.1 Detailed Description 336 7.119.1 Detailed Description 336 7.120/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/rotate_180_apu.b File Reference 336 7.121/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_act.cpp File Reference 336 7.121/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference 346 7.122/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference 346 7.122/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/	·	
sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_apu.h File Reference 33.7.114.1 Detailed Description 334.7.115/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_act.cpp File Reference 33.4.7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference 33.5.7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference 33.6.7.117/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_act.cpp File Reference 33.7.118/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp File Reference 33.7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h File Reference 33.7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_act.cpp File Reference 33.7.120/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference 34.0.7.121/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h File Reference 34.0.7.122.1 Detailed Description 34.0.7.122.1 Detailed Des	7.113.1 Detailed Description	333
7.115/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_acf.cpp File Reference 334 7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference 335 7.116/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_acf.cpp File Reference 336 7.118/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp File Reference 337 7.118.1 Detailed Description 337 7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h File Reference 338 7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.cpp File Reference 338 7.120/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference 338 7.121/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference 340 7.122.1 Detailed Description 340 7.123/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h File Reference 340 7.122.1 Detailed Description 340 7.123/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_act.cpp File Reference 340 7.124/home/bamboo/bam		
sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_acf.cpp File Reference	7.114.1 Detailed Description	334
sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference	· · · · · · · · · · · · · · · · · · ·	
7.117/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_acf.cpp File Reference	·	
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_acf.cpp File Reference	7.116.1 Detailed Description	336
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp File Reference	·	
7.119/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h File Reference		
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h File Reference	7.118.1 Detailed Description	337
7.120/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.cpp File Reference		
_sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.cpp File Reference	7.119.1 Detailed Description	338
_sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference	·	
7.122/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h File Reference 340 7.122.1 Detailed Description	·	
_sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h File Reference	7.121.1 Detailed Description	339
7.123/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_acf.cpp File Reference	·	
_sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_acf.cpp File Reference 340 7.124/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234← _sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.cpp File Reference 341 7.124.1 Detailed Description	7.122.1 Detailed Description	340
_sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.cpp File Reference	•	
7.125/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234	·	
•	7.124.1 Detailed Description	342
	·	

CONTENTS xxiii

7.125.1 Detailed Description	343
7.126/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_acf.cpp File Reference	
$7.127/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow \\ _sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.cpp\ File\ Reference$	
7.127.1 Detailed Description	344
7.128/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.h File Reference	
7.128.1 Detailed Description	345
$7.129/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow _sdk/kernels/apu/sample_optimization_kernels/src/sat_acf.cpp File Reference$	
7.130/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_optimization_kernels/src/sat_apu.cpp File Reference	
7.130.1 Detailed Description	346
7.131/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_optimization_kernels/src/sat_apu.h File Reference	
7.131.1 Detailed Description	347
7.132/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_acf.cpp File Reference	
7.133/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.cpp File Reference	
7.133.1 Detailed Description	348
7.134/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.h File Reference	
7.134.1 Detailed Description	349
7.135/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/downsample_acf.cpp File Reference	
7.136/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.cpp File Reference	
7.136.1 Detailed Description	351
7.137/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.h File Reference	
7.137.1 Detailed Description	352
7.138/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/upsample_acf.cpp File Reference	
7.139/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.cpp File Reference	
7.139.1 Detailed Description	352
7.140/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.h File Reference	
7.140.1 Detailed Description	353
7.141/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_statistics_kernels/src/accumulation_acf.cpp File Reference	
7.142/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.cpp File Reference	

xxiv CONTENTS

	7.142.1 Detailed Description	355
	7.143/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.h File Reference	
	7.143.1 Detailed Description	356
	7.144/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_statistics_kernels/src/column_sum_acf.cpp File Reference	
	7.145/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/sample_statistics_kernels/src/column_sum_apu.h File Reference	
	7.145.1 Detailed Description	357
	7.146/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_acf.cpp File Reference	
	7.147/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/sample_statistics_kernels/src/histofgrad_apu.h File Reference	
	7.147.1 Detailed Description	359
	7.148/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_statistics_kernels/src/reduction_acf.cpp File Reference	
	7.149/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/sample_statistics_kernels/src/reduction_apu.h File Reference	
	7.149.1 Detailed Description	361
	7.150/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_acf.cpp File Reference	
	7.151/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234- _sdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_apu.h File Reference	
	7.151.1 Detailed Description	363
	7.152/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-gsdk/kernels/apu/vfxp_math_lib/include/vfxp_math_inline.h File Reference	
Inc	dex	365

Chapter 1

Main Page

The herein provided kernel SDK is meant as a starting point for the implementation of computing kernels for the S32V234 APEX device. It has the purpose of being an inspiration source for the development of own kernels, since it touches on the different programming aspects of the APEX device.

Organization of the SDK:

The SDK has several component libraries:

- Arithmetic kernels: Provide basic operators for element-wise addition, subtraction, multiplication, division and arithmetic shifting
- **Comparison kernels**: Provide basic element-wise comparison operators like less than, less-than-or-equal, binary AND operator and binary descriptor matches
- · Conversion kernels: Conversion kernels from 16 to 8 bit and from RGB format to grayscale
- Display kernels: Gives examples of marking an image at certain points as overlay or in a certain color channel.
- Feature detection: Provides two corner detection algorithms FAST9 and Harris corner detection
- Filtering: Offers kernels for general purpose filtering, and also the most used filters like Gaussian filtering, gradient computation, non-maximum suppression and saturation
- **Geometry**: Provides geometric transformations, like rotations and bilinear interpolation and also a replacement for indirect inputs, called offset selection
- · Morphology: Example of a morphological dilation operator.
- Object detection: Two object detection algorithms: Haar cascade and LBP (local binary pattern) cascade
- Optical Flow: Kernels needed for the Optical flow algorithm
- Optimization : Implementation of the Integral Image (SAT) kernel and a SAT-based box filter.
- · Resizing: Provides downsampling and upsampling kernels (gives examples of size changes inside a filter)
- Statistics: Provides kernels for statistics computations, such as a Histogram kernel, a vector-to-scalar reduction kernel and an accumulation kernel.

The library's directory has following subcomponents:

- <category>_kernels/src: the directory containing the kernel source and header files of each library
- **BUILD.mk**: this file will be included by Makefiles which are used for target compilations, containing a list of all sources from the <category>_kernels/src subdirectory which shall be included into the library and all the necessary dependencies of these sources.

2 Main Page

 <category>_kernels/build-deskwin32/mvc : Contains Visual Studio projects which can be included into higher level application Visual Studio projects .

• <category>_kernels/build-<platform>-<compiler>-<OS>-d/ : Makefile builds for ACF/APEX Will contain build results for the corresponding platform.

Inside the <category>_kernels/src directory the files obey a certain naming convention which is as follows:

- apu_<KernelCategory>_impl.{c,h}: contain the kernel code which shall be executed on the APEX
- apu_<KernelCategory>.{c,h}: contain the ACF-kernel wrappers which provide the interface for the ACF-Graphs

Chapter 2

Module Index

2.1 Modules

Here is	a list	of all	modules:
1 1010 10	u	o. a	modaloo.

UserAPI				19
Apexcv_pro		 		20
Fast				21
Image_proc		 		23
Histogram_equalization				24
Arithmetic		 		26
Addition				27
Difference				
Division				44
Multiplication				51
Square				
Maximum				
Comparison		 		77
Comparison				78
Binary descriptor matching				95
Conversion		 		97
Bit width conversion				98
Color conversion				100
Display		 		106
Marking on images				
Feature detection				
Fast9 feature detection				
Harris corner detection				
Sum of Absolute Differences				
Filtering				
Col filter				
Correlation				
filtering				
Median filtering				
Sobel Filtering				
Gaussian filtering				
Image gradient				147
Non-maximum suppression				
Row_filter				
Saturation				155
Geometry		 		158

4 Module Index

Remap bilinear	9
Remap nearest	2
Rotate image by 180 deg	4
Norphology	6
Dilation	7
Object detection	9
Haar cascade object detection	
Dptimization	
Summed area table	
desizing	
Image downsampling	4
Image upsampling	7
tatistics	9
Accumulation	
Histofgrad	
of gradients	
Column sum	

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

APEX_HaarCascadeFeature)3
APEX_HaarCascadeStage)3
APEX_lbpFeature)3
APEX_lbpStage)4
point_t)4
rect_t)4
RESIZE_DESCRIPTOR	
Struct for Resize descriptor)4
resize_descriptor)6

6 Class Index

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_↔
sdk/kernels/apu/apexcv_gdc_ldw2/src/cg_kernel.h
Contains the prototypes for the APU kernels found in cg_kernel.a
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/apexcv_gdc_ldw2/src/ Kernels_Configuration.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←
sdk/kernels/apu/apexcv_gdc_ldw2/src/lane_detection_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_gdc_ldw2/src/linear_regression_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_gdc_ldw2/src/remap_bilinear_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_gdc_ldw2/src/thresh_val_histogram_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_gdc_ldw2/src/threshold_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ↔
sdk/kernels/apu/apexcv_pro_affine/src/ affine_definitions.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_acf.cpp
ACF metadata and wrapper function for the affine transformation
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_acf.h
ACF metadata and wrapper function for the affine transformation
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_apu.cpp
ACF Affine Transform Wrapper
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_apu.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←
sdk/kernels/apu/apexcv_pro_aggcf/src/aggcf_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_aggcf/src/ aggcf_apu.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/apexcv_pro_blockmatching/src/blockmatching_acf.h
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow
sdk/kernels/apu/apexcv_pro_blockmatching/src/blockmatching_apu.h
• • • • • • • • • • • • • • • • •

8 File Index

$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/apexcv_pro_brief/src/brief_acf.cpp	
ACF metadata and wrapper function for the BRIEF	211
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_brief/src/brief_acf.h	
ACF metadata for the BRIEF	212
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_brief/src/ brief_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_canny/src/canny_acf.cpp	04.4
ACF Metadata and wrapper function for Canny edge detector	214
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_canny/src/canny_acf.h	010
ACF Metadata for Canny edge detector	216
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_canny/src/canny_apu.cpp	218
Canny Edge Detection Kernels	210
sdk/kernels/apu/apexcv_pro_canny/src/canny_apu.h	
Canny Edge Detection Kernels	218
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow	210
sdk/kernels/apu/apexcv_pro_fast/src/fast_acf.cpp	220
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	220
sdk/kernels/apu/apexcv_pro_fast/src/fast_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	• •
sdk/kernels/apu/apexcv_pro_fast/src/fast_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	• •
sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.cpp	
ACF metadata and wrapper function for the Good Features To Track	221
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.h	
ACF metadata for the Good Features To Track	224
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_acf.cpp	
ACF metadata and wrapper function for the harris corner	226
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_acf.h	
ACF metadata for the harris corner	228
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_histogram_equalization/src/histogram_equalization_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/apexcv_pro_histogram_equalization/src/histogram_equalization_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/apexcv_pro_hog/src/ hog_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/apexcv_pro_hog/src/hog_apu.h	
HOG APU kernel implementation	230
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_hough/src/ cg_kernel.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/apexcv_pro_hough/src/ hough_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/apexcv_pro_hough/src/ hough_apu.h	??

4.1 File List

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_hough/src/hough_config_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.cpp	
ACF metadata and wrapper function for the Laplacian Pyramid	231
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ ↔	201
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.h	
	000
ACF metadata for the Laplacian Pyramid	233
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←	
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.cpp	
APU Laplacian Pyramid Implementation	234
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow$	
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.h	
APU Laplacian Pyramid Header	234
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_lbp/src/ lbp_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_lbp/src/ lbp_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ↔	•
sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.cpp	
=	236
··	230
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.h	
ACF metadata for the L-K optical flow pyramid	239
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_lkpyramid/src/ lkpyramid_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_setup.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_acf.cpp	
ACF metadata and wrapper function for the L-K optical flow tracker	241
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_acf.h	
ACF metadata for the L-K optical flow tracker	243
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ↔	
sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ ←	
sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_setup.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_orb/src/ orb_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/apexcv_pro_orb/src/ orb_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_acf.cpp	
ACF metadata and wrapper function for the image pyramid creation	245
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ↔	
sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_acf.h	
ACF metadata for the image pyramid creation	247
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ↔	
sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_apu.cpp	
Apu Image Pyramid Implementation	248
	240
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_apu.h	0.40
APU pyramid creation algorithm implementation	249
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/apexcv_pro_remap/src/cg_kernel.h	_
Contains the prototypes for the APU kernels found in cg_kernel.a	207

10 File Index

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_remap/src/ common_types.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_remap/src/ remap_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_remap/src/remap_apu.h	
Declaration of kernel functions for Remap	249
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_\leftarrow$	
sdk/kernels/apu/apexcv_pro_resize/src/cg_kernel.h	
Contains the prototypes for the APU kernels found in cg_kernel.a	207
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_resize/src/passthrough_kernel_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/apexcv_pro_resize/src/ resize_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_\leftarrow$	
sdk/kernels/apu/apexcv_pro_resize/src/ resize_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_\leftarrow$	
sdk/kernels/apu/apexcv_pro_resize/src/ resize_definitions.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_\leftarrow$	
sdk/kernels/apu/apexcv_pro_resize/src/resize_definitions_apu.h	
General definitions and declarations (Standard C), for Vertical Resizer	252
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_{\leftarrow}$	
sdk/kernels/apu/apexcv_pro_tmo/src/tmo_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_\leftarrow$	
sdk/kernels/apu/apexcv_pro_tmo/src/tmo_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 $_{\leftarrow}$	
sdk/kernels/apu/dnn/src/apexdnn_sn_acf.cpp	
ACF metadata and wrapper function for the Convolutional Neural Networks	254
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/dnn/src/apexdnn_sn_acf.h	
ACF metadata and wrapper function for the APEXDNN Library	256
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/dnn/src/ apexdnn_sn_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow$	
sdk/kernels/apu/dnn/src/neural_network_acf.cpp	
ACF metadata and wrapper function for the Convolutional Neural Networks	259
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/dnn/src/neural_network_acf.h	
ACF metadata and wrapper function for the Convolutional Neural Networks	263
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/dnn/src/ neural_network_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/add_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/add_apu.h	
Element-wise addition implementation for APEX	265
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/difference_acf.cpp	267
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/difference_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/difference_apu.h	
Computes the pixelwise difference btw. two images	268
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	000
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_acf.cpp	269
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_arithmetic_kernels/src/ dot_division_acf.h	??

4.1 File List

$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_apu.h	
Element-wise division of two vectors/matrices. If divisor is zero, a value of zero is returned (in	
order not to influence following arithmetics)	270
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_acf.cpp	272
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ color=1000000000000000000000000000000000000$	
sdk/kernels/apu/sample_arithmetic_kernels/src/ dot_multiplic_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_apu.h	
Element-wise multiplication of two vectors/matrices	274
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ color=1000000000000000000000000000000000000$	
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_sqr_acf.cpp	276
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_arithmetic_kernels/src/ dot_sqr_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_arithmetic_kernels/src/dot_sqr_apu.h	
Element-wise square of the input matrix	276
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ color=1000000000000000000000000000000000000$	
sdk/kernels/apu/sample_arithmetic_kernels/src/max_acf.cpp	277
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_arithmetic_kernels/src/ max_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_arithmetic_kernels/src/max_apu.h	
Element-wise maximum implementation for APEX	278
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.cpp	
ACF metadata and wrapper function for the harris corner	279
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.h	
ACF metadata and wrapper function for the harris corner	281
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_apu.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_comparison_kernels/src/comparison_acf.cpp	283
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_comparison_kernels/src/ comparison_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.cpp	
Compare functions implementation for APEX	285
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.h	
Element-wise comparison implementation for APEX	285
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_acf.cpp	287
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_comparison_kernels/src/ match_descriptors_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ color=1000000000000000000000000000000000000$	
sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.cpp	
Binary descriptor matching implementation for APEX	288
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.h	
Binary descriptor matching	289
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_acf.cpp	289
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_conversion_kernels/src/ cvt_16low_to_8_acf.h	??

12 File Index

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.cpp	000
16low_to_8 implementation for APEX	290
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	001
sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.h	291
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	291
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_acf.cpp	29
sdk/kernels/apu/sample_conversion_kernels/src/ rgb_to_grayscale_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.cpp	
Rgb_to_grayscale implementation for APEX	292
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	202
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.h	293
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_acf.cpp	293
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←	
sdk/kernels/apu/sample_conversion_kernels/src/ rgb_to_hsv_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.cpp	
Rgb_to_grayscale implementation for APEX	295
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.h	295
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/mark_acf.cpp	296
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/ mark_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/mark_apu.cpp	
Mark implementation for APEX	297
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_display_kernels/src/mark_apu.h	
Image marking implementation for APEX	297
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp	298
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp	200
Mark_color_channel implementation for APEX	299
sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.h	
Color channel image marking implementation for APEX	299
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	200
sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_acf.cpp	300
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_apu.cpp	
FAST 9 corner detection implementation for APEX	301
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_apu.h	
FAST 9 corner detection implementation for APEX	301
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_feature_detection_kernels/src/harris_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_feature_detection_kernels/src/harris_apu.h	??

4.1 File List

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_feature_detection_kernels/src/harris_defines.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_feature_detection_kernels/src/sad_acf.cpp	302
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample feature detection kernels/src/sad_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_feature_detection_kernels/src/sad_apu.h	
SAD - Sum of absolute Differences implementation for APEX	302
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	302
•	??
sdk/kernels/apu/sample_feature_detection_kernels/src/harris_complex/harris_acf.h	11
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	00
sdk/kernels/apu/sample_feature_detection_kernels/src/harris_complex/harris_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_acf.cpp	303
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_filtering_kernels/src/ col_filter_a_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.cpp	
Filtering with general filter image columns - implementation for APEX	304
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ ~	
sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.h	
A column filter implementation for the APU	305
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/correlation_acf.cpp	306
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/correlation_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_filtering_kernels/src/correlation_apu.cpp	
Convolution with general filter 1D/2D	308
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	300
· —	
sdk/kernels/apu/sample_filtering_kernels/src/correlation_apu.h Correlation with general filter 1D/2D	200
9	309
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	044
sdk/kernels/apu/sample_filtering_kernels/src/filter_a_acf.cpp	311
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/filter_a_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.cpp	
Filtering with general filter 1D/2D implementation for APEX	311
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.h	
Filtering with general filter 1D/2D	312
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ ~	
sdk/kernels/apu/sample_filtering_kernels/src/filter_a_cfg.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_acf.cpp	312
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.cpp	
3x3 Median filter implementation for APEX	313
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow	515
sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.h	
3x3 Median filter implementation for APEX	314
	J 14
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	04.4
sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_acf.cpp	314

14 File Index

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.cpp	
3x3 Sobel filter implementation for APEX	315
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.h	
3x3 Sobel filter implementation for APEX	316
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_acf.cpp	316
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	-00
sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_apu.h	017
Element-wise addition implementation for APEX	317
sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_acf.cpp	318
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	310
sdk/kernels/apu/sample_filtering_kernels/src/ gauss_3x3_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	• •
sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_apu.cpp	
3x3 Gauss filter	319
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	0.0
sdk/kernels/apu/sample filtering kernels/src/gauss 3x3 apu.h	
3x3 Gaussian filter implementation for APEX	320
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_acf.cpp	320
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.cpp	
5x5 Gauss filter implementation for APEX	321
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.h	
5x5 Gaussian filter implementation for APEX	322
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	
sdk/kernels/apu/sample_filtering_kernels/src/gradient_acf.cpp	322
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/ gradient_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.cpp	004
Gradient implementation for APEX	324
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.h Image gradient implementation for APEX	324
/home/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	324
sdk/kernels/apu/sample filtering kernels/src/nms acf.cpp	325
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	020
sdk/kernels/apu/sample_filtering_kernels/src/ nms_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	• •
sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.cpp	
Non-maximum supression filter implementation for APEX	326
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.h	
Non-maximum suppression implementation for APEX	326
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_acf.cpp	327

4.1 File List

$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow 1.00000000000000000000000000000000000$	
sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_acf.h	?
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_apu.cpp	
Filtering with general filter image rows - implementation for APEX	8
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_apu.h	
Row filter implementation for APU2	8
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_acf.cpp	9
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	_
sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_acf.h	?
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.cpp	_
Saturate_nonzero implementation for APEX	U
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.h	
Non-zero saturation implementation for APEX	U
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	•
sdk/kernels/apu/sample_filtering_kernels/src/symmetry_flags.h	•
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_geometry_kernels/src/disparity_acf.cpp	. !
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	_
sdk/kernels/apu/sample_geometry_kernels/src/disparity_acf.h	•
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_geometry_kernels/src/disparity_apu.h Element-wise addition implementation for APEX	
	_
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_acf.h	9
sdk/kernels/apu/sample_geometry_kernels/src/ remap_bilinear_acf.h	•
sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_apu.h	
Element-wise interpolation between pixels of an image for APEX	4
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_←	J
sdk/kernels/apu/sample geometry kernels/src/remap nearest acf.cpp	4
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_geometry_kernels/src/ remap_nearest_acf.h	?
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	•
sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h	
Element-wise interpolation between pixels of an image for APEX	5
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←	Ĭ
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_acf.cpp	6
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←	Ī
sdk/kernels/apu/sample_geometry_kernels/src/ rotate_180_acf.h	?
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp	
Rotate an image by 180 deg implementation for APEX	7
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h	
180-degree rotation implementation for APEX	8
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 ←	
sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.cpp	8
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.h	?
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp	
Dilate_diamond implementation for APEX	9

16 File Index

$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h	
Image diamond dilation implementation for APEX	340
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_object_detection_kernels/src/ cascade_definitions.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_acf.cpp	340
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.cpp	
Haar cascade implementation for APEX	341
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.h	342
Object detection based on Haar-like features implementation for APEX	342
sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_acf.cpp	343
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	343
sdk/kernels/apu/sample object detection kernels/src/ lbp_cascade_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.cpp	
LBP cascade implementation for APEX	344
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	• • • • • • • • • • • • • • • • • • • •
sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.h	
Object detection based on LBP features implementation for APEX	344
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftrightarrow	
sdk/kernels/apu/sample_optimization_kernels/src/sat_acf.cpp	346
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_optimization_kernels/src/sat_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_optimization_kernels/src/sat_apu.cpp	
Sat_box_filter implementation for APEX	346
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \leftarrow \\$	
sdk/kernels/apu/sample_optimization_kernels/src/sat_apu.h	
Summed area table implementation for APEX	347
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_acf.cpp	347
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.cpp	348
Sat_box_filter implementation for APEX	340
sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.h	
Box filter using SAT implementation for APEX	349
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 \leftarrow	040
sdk/kernels/apu/sample_resizing_kernels/src/downsample_acf.cpp	350
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	000
sdk/kernels/apu/sample_resizing_kernels/src/downsample_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	••
sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.cpp	
Downsample implementation for APEX	350
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.h	
Image downsample implementation for APEX	351
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_resizing_kernels/src/resize_definitions.h	
General definitions and declarations (Standard C), for Vertical Resizer	250

4.1 File List

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_resizing_kernels/src/upsample_acf.cpp	352
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_resizing_kernels/src/ upsample_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.cpp	
Upsample implementation for APEX	352
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.h	
Image upsample implementation for APEX	353
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/accumulation_acf.cpp	354
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/ accumulation_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.cpp	
Builds the sum of all elements of a chunk and writes out a vector of sum values	355
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.h	
Accumulation of values from a chunk	355
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/column_sum_acf.cpp	356
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/ column_sum_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/column_sum_apu.h	
Cumputing sum of columns of image for APEX ldw_v2 demo	357
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_acf.cpp	357
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow \\$	
sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_acf.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_apu.h	
Histogram of gradients computation for APEX	358
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_defines.h	??
$/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ \hookleftarrow 100000000000000000000000000000000000$	
sdk/kernels/apu/sample_statistics_kernels/src/histogram_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_statistics_kernels/src/histogram_apu.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	0.50
sdk/kernels/apu/sample_statistics_kernels/src/reduction_acf.cpp	359
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_statistics_kernels/src/ reduction_acf.h	??
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_statistics_kernels/src/reduction_apu.h	000
Reduction of a vector/image implementation for APEX	360
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	004
sdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_acf.cpp	361
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	??
sdk/kernels/apu/sample_statistics_kernels/src/ reduction_for_clmn_acf.h	"
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	
sdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_apu.h	362
Reduction of a vector/image implementation for APEX ldw_v2 demo	302
sdk/kernels/apu/vfxp_math_lib/include/vfxp_math_inline.h	363
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_	303
sdk/kernels/apu/vfxp_math_lib/include/ vfxp_types.h	??
survivernets/apu/vixp_matit_mo/molude/vixp_types.ii	: :

18 File Index

Chapter 5

Module Documentation

5.1 UserAPI

5.1.1 Detailed Description

This is the group of enum, structure and functions needs to be exposed to APEX MATH library user

5.2 Apexcv_pro

5.3 Fast 21

5.3 Fast

5.3.1 Detailed Description

Histogram.

Functions

- KERNEL_INFO fast_offset (" fast_offset ", 3, __port(__index(0), __identifier("OUTPUT_OFFSETS"), __
 attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __
 e0_size(1, 1), __ek_size(16, 1)), __port(__index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_V
 EC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__
 index(2), __identifier("CIRCUMFERENCE"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial
 __dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO **fast** (" fast ", 4, __port(__index(0), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_ ← VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(← index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_← data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("THRESHOLD"), __ attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("OFFSET_TABLE"), __attributes(ACF_ATTR_← SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(16, 1)))
- KERNEL_INFO fast_serialized (" fast_serialized ", 6, __port(__index(0), __identifier("INPUT"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_size(1, 1), _
 _ek_size(1, 1)), __port(__index(1), __identifier("THRESHOLD"), __attributes(ACF_ATTR_SCL_IN__
 STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)),
 _port(__index(2), __identifier("OFFSET_TABLE"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __
 _spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(16, 1)), __port(__index(3),
 _identifier("OUT_PACKED"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0,
 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(8192, 1)), __port(__index(4), __identifier("COUNTE_\times R"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s),
 _e0_size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("OUT_MAX_SIZE"), __attributes(ACF_\times ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1),
 __ek_size(1, 1)))
- KERNEL_INFO fast_nms (" fast_nms ", 4, __port(__index(0), __identifier("OUTPUT_0"), __attributes(AC← F_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), ← __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("THRESHOLD"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __← e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OFFSET_TABLE"), __attributes(ACF_AT← TR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_← size(16, 1)))
- KERNEL_INFO nms3x3 (" nms3x3 ", 2, __port(__index(0), __identifier("input"), __attributes(ACF_ATTR_
 VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_
 index(1), __identifier("output"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data
 _type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

5.3.2 Function Documentation

5.3.2.1 KERNEL_INFO nms3x3 (" nms3x3 " , 2 , __port(__index(0), __identifier("input"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("output"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Non-maximum suppression 3x3 kernel metadata.

2	Number of ports
Port	Define for name of input image (unsigned 8bit)
NMS16_KN_IN	
Port NMS16_K↔	Define for name of output image (unsigned 8bit)
N_OUT	

5.4 Image_proc 23

5.4 Image_proc

5.4.1 Detailed Description

Collaboration diagram for Image_proc:



Modules

• Histogram_equalization Histogram.

5.5 Histogram equalization

5.5.1 Detailed Description

Histogram.

Collaboration diagram for Histogram equalization:



Functions

- KERNEL_INFO apu_histogram_equalization (" apu_histogram_equalization ", 4, __port(__index(0), _
 _ identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_0"), __attributes(According CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("LUT_IN"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __coloretic coloretic co
- KERNEL_INFO apu_generate_lut (" apu_generate_lut ", 3, __port(__index(0), __identifier("LUT_OUT"), _
 _ attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __
 e0_size(1, 1), __ek_size(256, 1)), __port(__index(1), __identifier("IMAGE_HISTOGRAM"), __attributes(A
 CF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1),
 __ek_size(256, 1)), __port(__index(2), __identifier("NUM_PIXELS"), __attributes(ACF_ATTR_SCL_IN_ST
 ATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

5.5.2 Function Documentation

```
5.5.2.1 KERNEL_INFO apu_generate_lut ( " apu_generate_lut " , 3 , __port(_index(0), __identifier("LUT_OUT"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(256, 1)) , __port(_index(1), __identifier("IMAGE_HISTOGRAM"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)) , __port(_index(2), __identifier("NUM_PIXELS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Histogram_equalization kernel metadata.

apu_equalized⇔	Kernel name
_lut	
3	Number of ports
Port IMAGE_H↔	input image histogram (unsigned 32bit)
ISTOGRAM	

Port LUT	histogram equalization look up table, CU count X 256 (unsigned 8bit)
Port	number of pixels in input image
NUM_PIXELS	

```
5.5.2.2 KERNEL_INFO apu_histogram_equalization ( " apu_histogram_equalization " , 4 , __port(_index(0), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("LUT_IN"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(256, 1)) , __port(__index(3), __identifier("VECTORIZED_LUT_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(256, 1)) )
```

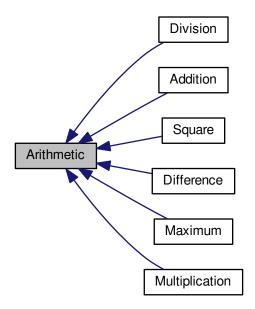
Histogram_equalization kernel metadata.

apu_histogram⇔	Kernel name
_equalization	
2	Number of ports
Port INPUT_0	input image (unsigned 8bit)
Port OUTPUT_0	histogram equalization vector output, CU count X 256 (unsigned 8bit)
Port LUT	histogram equalization look up table, CU count X 256 (unsigned 8bit)

5.6 Arithmetic

5.6.1 Detailed Description

Collaboration diagram for Arithmetic:



Modules

Addition

Element-wise addition.

• Difference

Element-wise difference.

• Division

Element-wise division.

Multiplication

Element-wise multiplication.

Square

Element-wise square.

Maximum

Element-wise maximum.

5.7 Addition 27

5.7 Addition

5.7.1 Detailed Description

Element-wise addition.

Collaboration diagram for Addition:



Functions

- KERNEL_INFO apu_add_in16s_out32s (" apu_add_in16s_out32s ", 3, __port(_index(0), __identifier("I← NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_← size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), ← identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_← type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_add_in32s_out32s (" apu_add_in32s_out32s ", 3, __port(_index(0), __identifier("I← NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_← size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), ← identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_← type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_add_in32Q3_28_out32Q3_28 (" apu_add_in32Q3_28_out32Q3_28 ", 4, __port(__ index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(Actributes(Actributes(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __ek_size(1, 1)))
- KERNEL_INFO apu_add_in64s_out64s ("apu_add_in64s_out64s", 6, __port(__index(0), __identifier("IN PUT_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __
 e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_0_LOW"), __attributes(ACF_ATT Portion R_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(Put_0_index(2), __identifier("INPUT_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), Put_0_index(3), __identifier("INPUT_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("OUTPUT_0_HIGH"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(5), __port(_index(5), __port(_por

void add (vec16u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)
 Elementwise unsigned 8bit addition => unsigned 16bit.

void add_in16s_out32s (vec32s *dst, vec16s *srcImage0, vec16s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 16bit addition => signed 32bit.

void add_in32s_out32s (vec32s *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit addition => signed 32bit.

void add_in32u_out32u (vec32u *dst, vec32u *srcImage0, vec32u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 32bit addition => unsigned 32bit.

void add_in64s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcImage0_high, vec32u *srcImage0←
 _low, vec32s *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit addition => signed 64bit.

- void add_in64u_out64u (vec32u *dst_high, vec32u *dst_low, vec32u *srcImage0_high, vec32u *srcFmage0_low, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)
 Elementwise unsigned 64bit addition => unsigned 64bit.
- void add_in32Q3_28_out32Q3_28 (vec32s *dstInt, vec32s *dstFrac, vec32s *srcImage0, vec32s *src
 Image1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise 32bit Q3.28 addition => 64bit in [Q3.28] (integer part) and {Q3.28} (fractional part) format.

- KERNEL_INFO apu_gauss_3x1 (" apu_gauss_3x1 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void gauss_3x1 (vec08u *dst, vec08u *srcImage0, int bw, int bh, int inStrideW, int outStrideW)
 Elementwise unsigned 8bit addition => unsigned 16bit.
- KERNEL_INFO apu_disparity (" apu_disparity ", 3, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 64, 1, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT
 PUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_
 size(1, 1), __ek_size(1, 1)))
- void disparity (vec08u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int cw, int ch, int in← StrideW0, int inStrideW1, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

5.7.2 Function Documentation

5.7.2.1 void add (vec16u * dst, vec08u * srcImage0, vec08u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

Addition btw two unsigned 8bit matrices with 16bit result: dst[i] = srcImage0[i] + srcImage1[i]

dst	- [Output] 16bit destination block pointer
srcImage0	- [Input] 8bit source block pointer of img 0
srcImage1	- [Input] 8bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height

5.7 Addition 29

inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.2 void add_in16s_out32s (vec32s * dst, vec16s * srcImage0, vec16s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 16bit addition => signed 32bit.

Addition btw two signed 16bit matrices with 32bit result: dst[i] = srcImage0[i] + srcImage1[i]

Parameters

dst	- [Output] 32bit destination block pointer
srcImage0	- [Input] 16bit source block pointer of img 0
srcImage1	- [Input] 16bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.3 void add_in32Q3_28_out32Q3_28 (vec32s * dstInt, vec32s * dstFrac, vec32s * srcImage0, vec32s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise 32bit Q3.28 addition => 64bit in [Q3.28] (integer part) and {Q3.28} (fractional part) format.

Addition btw two 32bit fixed point numbers in Q3.28 format: dstInt[i] = int(srcImage0[i] + srcImage1[i]); dstFrac[i] = frac(srcImage0[i] + srcImage1[i]); (the number below zero, which is represented here as $0.n * 2^{2}$

Parameters

dstInt	- [Output] 32bit the integer part of the addition
dstFrac	- [Output] 32bit the fractional part of the addition in Q3.28 fixed point format
srcImage0	- [Input] 32bit source block pointer of vector 0 in Q3.28 fixed point format
srcImage1	- [Input] 32bit source block pointer of vector 1 in Q3.28 fixed point format
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.4 void add_in32s_out32s (vec32s * srcImage0, vec32s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit addition => signed 32bit.

Addition btw two signed 32bit matrices: dst[i] = srcImage0[i] + srcImage1[i]

Warning

No out of range above 32bits is checked

dst	- [Output] signed 32bit destination block pointer

srcImage0	- [Input] signed 32bit source block pointer of img 0
srcImage1	- [Input] signed 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.5 void add_in32u_out32u (vec32u * srcImage0, vec32u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 32bit addition => unsigned 32bit.

Addition btw two unsigned 32bit matrices: dst[i] = srcImage0[i] + srcImage1[i]

Warning

No out of range above 32bits is checked

Parameters

dst	- [Output] unsigned 32bit destination block pointer
srcImage0	- [Input] unsigned 32bit source block pointer of img 0
srcImage1	- [Input] unsigned 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.6 void add_in64s_out64s (vec32s * dst_high, vec32u * dst_low, vec32s * srcImage0_high, vec32u * srcImage0_low, vec32s * srcImage1_high, vec32u * srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit addition => signed 64bit.

Addition btw two signed 64bit matrices: dst[i] = srcImage0[i] + srcImage1[i]

Warning

No out of range above 64bits is checked

Parameters

dst_high	- [Output] High word 32bit destination block pointer
dst_low	- [Output] Low word 32bit destination block pointer
srcImage0_high	- [Input] High word 32bit source block pointer of img 0
srcImage0_low	- [Input] Low word 32bit source block pointer of img 0
srcImage1_high	- [Input] High word 32bit source block pointer of img 1
srcImage1_low	- [Input] Low word 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.7 void add_in64u_out64u (vec32u * dst_high, vec32u * dst_low, vec32u * srcImage0_high, vec32u * srcImage0_low, vec32u * srcImage1_high, vec32u * srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 64bit addition => unsigned 64bit.

5.7 Addition 31

Addition btw two unsigned 64bit matrices: dst[i] = srcImage0[i] + srcImage1[i]

Warning

No out of range above 64bits is checked

Parameters

dst_high	- [Output] High word 32bit destination block pointer
dst_low	- [Output] Low word 32bit destination block pointer
srcImage0_high	- [Input] High word 32bit source block pointer of img 0
srcImage0_low	- [Input] Low word 32bit source block pointer of img 0
srcImage1_high	- [Input] High word 32bit source block pointer of img 1
srcImage1_low	- [Input] Low word 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

```
5.7.2.8 KERNEL_INFO apu_add ( "apu_add ", 3, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two unsigned 8bit images. Outputs 16bit unsigned addition result.

Parameters

ADD_KN	Define for Kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
ADD_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
ADD_KN_INB	
Port	Define for name of addition result of the two images (unsigned 16bit).
ADD_KN_OUT	

```
5.7.2.9 KERNEL_INFO apu_add_in16s_out32s ( "apu_add_in16s_out32s ", 3 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two signed 16bit images. Outputs signed 32bit addition result

	ADD_In16s_←	Define for Kernel name
	Out32s_KN	
Ī	3	Number of ports
	Port	Define for name of first input image (signed 16bit)
	ADD KN INA	

Port	Define for name of second input image (signed 16bit)
ADD_KN_INB	
Port	Define for name of addition result (signed 32bit).
ADD_KN_OUT	

```
5.7.2.10 KERNEL_INFO apu_add_in32Q3_28_out32Q3_28 ( "apu_add_in32Q3_28_out32Q3_28 ", 4, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("OUTPUT_Frac"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
```

Addition kernel metadata. Adds pixelwise two images in fixed point format of the type Q3.28. Outputs signed 32bit the integer addition result (in normal integer format) and the fractional addition result in Q3.28 format

Warning

Does not check out of range values above/below +/-2\(^(31-1))

Parameters

<i>ADD_32S_Q3</i> ⇔	Define for Kernel name
_28_KN	
4	Number of ports
Port	Define for name of first input image (signed 32bit) in fixed point Q3.28 format
ADD_KN_INA	
Port	Define for name of second input image (signed 32bit) in fixed point Q3.28 format
ADD_KN_INB	
Port	Define for name of 32bit signed integer part of addition result of the two images.
ADD_KN_OUT	
Port ADD_KN_←	Define for name of 32bit fractional part of addition result of the two images in fixed point Q3.28
OUT_FRAC	format.

```
5.7.2.11 KERNEL_INFO apu_add_in32s_out32s ( "apu_add_in32s_out32s " , 3 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two signed 32bit images. Outputs signed 32bit addition result

Warning

Does not check out of range values above/below +/-2^(31-1)

ADD_In32s_←	Define for Kernel name
Out32s_KN	
3	Number of ports
Port	Define for name of first input image (signed 32bit)
ADD_KN_INA	
Port	Define for name of second input image (signed 32bit)
ADD_KN_INB	
Port	Define for name of addition result (signed 32bit).
ADD_KN_OUT	

5.7 Addition 33

```
5.7.2.12 KERNEL_INFO apu_add_in64s_out64s ( " apu_add_in64s_out64s " , 6 , __port(_index(0), __identifier("INPUT_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_0_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("INPUT_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("INPUT_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(4), __identifier("OUTPUT_0_HIGH"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(5), __identifier("OUTPUT_0_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two signed 64bit images. Outputs signed 64bit addition result

Warning

Does not check out of range values above 2⁽⁶³⁻¹⁾

Parameters

ADD_In64s_←	Define for Kernel name
Out64s_KN	
6	Number of ports
Port ADD_KN_←	Define for name of signed 32bit high word of first input image
INA_HIGH	
Port ADD_KN_←	Define for name of unsigned 32bit low word of firstinput image
INA_LOW	
Port ADD_KN_←	Define for name of signed 32bit high word of second input image
INB_HIGH	
Port ADD_KN_←	Define for name of unsigned 32bit low word of second input image
INB_LOW	
Port ADD_KN_←	Define for name of signed 32bit high word of addition result
OUT_HIGH	
Port ADD_KN_←	Define for name of unsigned 32bit low word of addition result
OUT_LOW	

```
5.7.2.13 KERNEL_INFO apu_disparity ( " apu_disparity " , 3 , __port(__index(0), __identifier("INPUT_0"),  
__attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 64, 1, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two unsigned 8bit images. Outputs 16bit unsigned addition result.

ADD_KN	Define for Kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
ADD_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
ADD_KN_INB	
Port	Define for name of addition result of the two images (unsigned 16bit).
ADD_KN_OUT	

```
5.7.2.14 KERNEL_INFO apu_gauss_3x1 ( " apu_gauss_3x1 " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Addition kernel metadata. Adds pixelwise two unsigned 8bit images. Outputs 16bit unsigned addition result.

5.7 Addition 35

Parameters

ADD_KN	Define for Kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
ADD_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
ADD_KN_INB	
Port	Define for name of addition result of the two images (unsigned 16bit).
ADD_KN_OUT	

5.7.2.15 void disparity (vec08u * dst, vec08u * srcImage0, vec08u * srcImage1, int bw, int bh, int cw, int ch, int inStrideW0, int inStrideW1, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

Addition btw two unsigned 8bit matrices with 16bit result: dst[i] = srcImage0[i] + srcImage1[i]

Parameters

dst	- [Output] 16bit destination block pointer
srcImage0	- [Input] 8bit source block pointer of img 0
srcImage1	- [Input] 8bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.7.2.16 void gauss_3x1 (vec08u * dst, vec08u * srcImage0, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

Addition btw two unsigned 8bit matrices with 16bit result: dst[i] = srcImage0[i] + srcImage1[i]

dst	- [Output] 16bit destination block pointer
srcImage0	- [Input] 8bit source block pointer of img 0
srcImage1	- [Input] 8bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.8 Difference

5.8.1 Detailed Description

Element-wise difference.

Collaboration diagram for Difference:



Functions

- KERNEL_INFO apu_diff_in08u_out16s (" apu_diff_in08u_out16s ", 3, __port(_index(0), __identifier("I↔ NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0↔ __size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC↔ __IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ↔ index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_ ↔ data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in16s_out16s (" apu_diff_in16s_out16s ", 3, __port(__index(0), __identifier("I↔ NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0↔ __size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC↔ __IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__↔ index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_↔ data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in16s_out32s (" apu_diff_in16s_out32s ", 3, __port(__index(0), __identifier("I← NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0← __size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC← __IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0← data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in32s_out32s (" apu_diff_in32s_out32s ", 3, __port(__index(0), __identifier("I← NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC← __IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in64s_out64s (" apu_diff_in64s_out64s ", 6, __port(__index(0), __identifier("IN←PUT_A_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), _←e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_A_LOW"), __attributes(ACF_A←TTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INPUT_B_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("INPUT_B_←LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("OUTPUT_HIGH"), __attributes(ACF_ATTR_VEC_OU←T), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("OUTPUT_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1))
- void difference_filter_in08u_out16s (vec16s *dst, vec08u *srcA, vec08u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

5.8 Difference 37

Pixel-wise difference between two unsigned 8bit images => signed 16bit.

void difference_filter_in16s_out16s (vec16s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 16bit.

void difference_filter_in16s_out32s (vec32s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

void difference_filter_in32s_out32s (vec32s *dst, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

void difference_filter_in32u_out32s (vec32s *dst, vec32u *srcA, vec32u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 32bit images => signed 32bit.

void difference_filter_in32s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 32bit images => signed 64bit.

void difference_filter_in64s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStride
 Width)

Pixel-wise difference between two signed 64bit images => signed 64bit.

void difference_filter_in64u_out64s (vec32s *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStride
 Width)

Pixel-wise difference between two unsigned 64bit images => signed 64bit.

5.8.2 Function Documentation

```
5.8.2.1 KERNEL_INFO apu_diff_in08u_out16s ( "apu_diff_in08u_out16s ", 3, __port(_index(0), __identifier("INPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Difference kernel metadata. Computes pixelwise the difference between two unsigned 8bit images. Outputs 16bit signed difference result.

DIFF_In08u_←	Macro Definition for Kernel name
Out16s_KN	
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
DIFF_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
DIFF_KN_INB	
Port	Define for name of difference result of the two images (signed 16bit).
DIFF KN OUT	

```
5.8.2.2 KERNEL_INFO apu_diff_in16s_out16s ( " apu_diff_in16s_out16s " , 3 , __port(__index(0), __identifier("INPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Difference kernel metadata. Computes pixelwise the difference between two signed 16bit images. Outputs 16bit signed difference result.

Warning

No sanity checks for out of bounds values above/below $\pm -2^{\land}(15-1)$ are performed.

Parameters

DIFF_In16s_←	Macro Definition for Kernel name
Out16s_KN	
3	Number of ports
Port	Define for name of first input image (signed 16bit)
DIFF_KN_INA	
Port	Define for name of second input image (signed 16bit)
DIFF_KN_INB	
Port	Define for name of result of the two images (signed 16bit).
DIFF_KN_OUT	

```
5.8.2.3 KERNEL_INFO apu_diff_in16s_out32s ( "apu_diff_in16s_out32s ", 3, __port(_index(0), __identifier("INPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Difference kernel metadata. Computes pixelwise the difference between two signed 16bit images. Outputs 32bit signed difference result.

Parameters

DIFF_In16s_←	Macro Definition for Kernel name
Out32s_KN	
3	Number of ports
Port	Define for name of first input image (signed 16bit)
DIFF_KN_INA	
Port	Define for name of second input image (signed 16bit)
DIFF_KN_INB	
Port	Define for name of result of the two images (signed 32bit).
DIFF_KN_OUT	

```
5.8.2.4 KERNEL_INFO apu_diff_in32s_out32s ( " apu_diff_in32s_out32s " , 3 , __port(__index(0), __identifier("INPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Difference kernel metadata. Computes pixelwise the difference between two signed 32bit images. Outputs 32bit signed difference result.

5.8 Difference 39

Warning

No sanity checks for out of bounds values above/below $\pm -2^{(31-1)}$ are performed.

Parameters

DIFF_In32s_←	Macro Definition for Kernel name
Out32s_KN	
3	Number of ports
Port	Define for name of first input image (signed 32bit)
DIFF_KN_INA	
Port	Define for name of second input image (signed 32bit)
DIFF_KN_INB	
Port	Define for name of result of the two images (signed 32bit).
DIFF_KN_OUT	

```
5.8.2.5 KERNEL_INFO apu_diff_in64s_out64s ( "apu_diff_in64s_out64s " , 6 , __port(__index(0), __identifier("INPUT_A_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_A_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("INPUT_B_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("INPUT_B_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(4), __identifier("OUTPUT_HIGH"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(5), __identifier("OUTPUT_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Difference kernel metadata. Computes pixelwise the difference between two signed 64bit images. Outputs 64bit signed difference result.

Warning

No sanity checks for out of bounds values above/below +/- 2^{\wedge} (63-1) are performed.

Parameters

<i>DIFF_In64s_</i> ←	Macro Definition for Kernel name
Out64s_KN	
6	Number of ports
Port DIFF_KN↔	Define for name of first signed 32bit high word of input image
_INA_HIGH	
Port DIFF_KN↔	Define for name of first unsigned 32bit low word of input image
_INA_LOW	
Port DIFF_KN↔	Define for name of first signed 32bit high word of input image
_INB_HIGH	
Port DIFF_KN↔	Define for name of first unsigned 32bit low word of input image
_INB_LOW	
Port DIFF_KN↔	Define for name of signed 32bit high word of difference result
_OUT_HIGH	
Port DIFF_KN↔	Define for name of unsigned 32bit low word of difference result
_OUT_LOW	

5.8.2.6 void difference_filter_in08u_out16s (vec16s * dst, vec08u * srcA, vec08u * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 8bit images => signed 16bit.

Pixel-wise difference between two unsigned 8bit images with signed 16bit result. dst[i] = srcImageA[i] - srcImageB[i];

Parameters

dst	- [Output] 16bit Destination block pointer
srcA	- [Input] 8bit Source block pointer of img A
srcB	- [Input] 8bit Source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.7 void difference_filter_in16s_out16s (vec16s * dst, vec16s * srcA, vec16s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 16bit.

Pixel-wise difference between two signed 16bit images with 16bit result.

Warning

No out of range is checked

Parameters

dst	- [Output] signed 16bit Destination block pointer
srcA	- [Input] signed 16bit Source block pointer of img A
srcB	- [Input] signed 16bit Source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.8 void difference_filter_in16s_out32s (vec32s * dst, vec16s * srcA, vec16s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

Pixel-wise difference between two signed 16bit images with signed 32bit result

Parameters

dst	- [Output] signed 32bit Destination block pointer
srcA	- [Input] signed 16bit Source block pointer of img A
srcB	- [Input] signed 16bit Source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.9 void difference_filter_in32s_out32s (vec32s * dst, vec32s * srcA, vec32s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

Pixel-wise difference between two signed 32bit images with signed 32bit result

5.8 Difference 41

Warning

No out of range is checked!

Parameters

dst	- [Output] signed 32bit Destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
srcB	- [Input] signed 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.10 void difference_filter_in32s_out64s (vec32s * dst_high, vec32u * dst_low, vec32s * srcA, vec32s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 32bit images => signed 64bit.

Pixel-wise difference between two signed 32bit images with signed 64 bit result

Parameters

dst_high	- [Output] signed 32bit high word of destination block pointer
dst_low	- [Output] unsigned 32bit low word of destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
srcB	- [Input] signed 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.11 void difference_filter_in32u_out32s (vec32s * dst, vec32u * srcA, vec32u * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 32bit images => signed 32bit.

Pixel-wise difference between two unsigned 32bit images with signed 32bit result.

Warning

No out of range is checked. If the unsigned inputs are higher than 0x7fff ffff, wrong results will arise

Parameters

dst	- [Output] 32bit high word of destination block pointer
srcA	- [Input] unsigned 32bit source block pointer of img A
srcB	- [Input] unsigned 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.12 void difference_filter_in64s_out64s (vec32s * dst_high, vec32u * dst_low, vec32s * srcA_high, vec32u * srcA_low, vec32s * srcB_high, vec32u * srcB_low, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 64bit images => signed 64bit.

42 **Module Documentation** Pixel-wise difference between two signed 64bit images, given as low and high word blocks with signed 64bit result

5.8 Difference 43

Parameters

dst_high	- [Output] signed 32bit high word of destination block pointer
dst_low	- [Output] unsigned 32bit low word of destination block pointer
srcA_high	- [Input] signed 32bit high word source block pointer of img A
srcA_low	- [Input] unsigned 32bit low word source block pointer of img A
srcB_high	- [Input] signed 32bit high word source block pointer of img B
srcB_low	- [Input] unsigned 32bit low word source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.8.2.13 void difference_filter_in64u_out64s (vec32s * dst_high, vec32u * dst_low, vec32u * srcA_high, vec32u * srcA_low, vec32u * srcB_high, vec32u * srcB_low, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 64bit images => signed 64bit.

Pixel-wise difference between two unsigned 64bit images, given as low and high word blocks with signed 64bit result Parameters

dst_high	- [Output] signed 32bit high word of destination block pointer
dst_low	- [Output] unsigned 32bit low word of destination block pointer
srcA_high	- [Input] unsigned 32bit high word source block pointer of img A
srcA_low	- [Input] unsigned 32bit low word source block pointer of img A
srcB_high	- [Input] unsigned 32bit high word source block pointer of img B
srcB_low	- [Input] unsigned 32bit low word source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Output block width (in elements not bytes) including padding

5.9 Division

5.9.1 Detailed Description

Element-wise division.

Collaboration diagram for Division:



Functions

- KERNEL_INFO apu_dot_division (" apu_dot_division ", 3, __port(__index(0), __identifier("INPUT_A"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPU
 T"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_log2 (" apu_dot_log2 ", 2, __port(_index(0), __identifier("Log2_input"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek
 _size(1, 1)), __port(_index(1), __identifier("Log2_fact"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_division_N64s_D32s_Q64s (" apu_dot_division_N64s_D32s_Q64s ", 6, __port(← __index(0), __identifier("Div_N64s_NOM_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("Div_N64s_N← OM_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("Div_N64s_DIVISOR"), __attributes(ACF_ATT← R_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__ ← index(3), __identifier("Div_N64s_OUT_HIGH"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("Div_N64s_O← UT_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("Div_N64s_OUT_REM"), __attributes(ACF_AT← TR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1))
- void dot_division_filter (vec32s *res, vec32s *numerator, vec32s *denominator, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise division of signed 32bit integers.

void computeLog2 (vec08u *log2Fact, vec32s *input, int16s bw, int16s bh, int16s inStrideWidth, int16s out
 StrideWidth)

res[i] = log2(abs(input[i])) + 1, where input is a signed 32bit integer

void computeLog2u (vec08u *log2Fact, vec32u *input, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

5.9 Division 45

res[i] = log2(input[i]) + 1, where input is a unsigned 32bit integer

void compute64bitLog2 (vec08u *log2Fact, vec32s *input_high, vec32u *input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

```
res[i] = log2( abs(input[i]) ) + 1, where input is a signed 64bit integer
```

void compute64bitLog2u (vec08u *log2Fact, vec32u *input_high, vec32u *input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

```
res[i] = log2(input[i]) + 1, where input is a unsigned 64bit integer
```

void computeInv_NewtonRaphson (vec32s *invDiv, vec32s *div, vec08u *log2Fact, const int08s shiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise inverse of a signed integer vector using the NewtonRaphson algorithm.

void dot_division_filter_N64s_D32s_Q64s (vec32s *dst_high, vec32u *dst_low, vec32u *dst_rem, vec32s *nom_high, vec32u *nom_low, vec32s *divisor, int16s bw, int16s inStrideWidth, int16s outStride Width)

Elementwise division of a 64bit nominator by a 32bit divisor.

5.9.2 Function Documentation

```
5.9.2.1 KERNEL_INFO apu_dot_division ( "apu_dot_division ", 3 , __port(__index(0), __identifier("INPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Integer division kernel metadata. Divides pixelwise two signed 32bit images. Outputs signed 32bit integer division result.

Warning

Division by zero returns zero and not an out of range number.

Parameters

DIV_In32s_←	Define for Kernel name
Out_32s_KN	
3	Number of ports
Port	Define for name of first input image (signed 32bit).
DIV_KN_INA	
Port	Define for name of second input image (signed 32bit).
DIV_KN_INB	
Port	Define for name of division result (signed 32bit).
DIV_KN_OUT	

```
5.9.2.2 KERNEL_INFO apu_dot_division_N64s_D32s_Q64s ( " apu_dot_division_N64s_D32s_Q64s " , 6 ,
    __port(__index(0), __identifier("Div_N64s_NOM_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0),
    __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("Div_N64s_NOM_LOW"),
    __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1))
    , __port(__index(2), __identifier("Div_N64s_DIVISOR"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0),
    __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("Div_N64s_OUT_HIGH"),
    __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1))
    , __port(__index(4), __identifier("Div_N64s_OUT_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
    __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(5), __identifier("Div_N64s_OUT_REM"),
    __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Integer division kernel metadata. Divides pixelwise a 64bit integer image with a 32bit integer. Outputs signed 64bit integer division result.

Warning

Division by zero returns zero and not an out of range number.

Parameters

DIV_N64s_←	Define for Kernel name
D32s_Q64s_K	
6	Number of ports
Port DIV_N64s↔	Define for name of signed 32bit high word of input image (i.e. the nominator)
_ <i>D32s_Q64s_</i> ←	
KN_NOM_HIGH	
Port DIV_N64s↔	Define for name of unsigned 32bit low word of input image (i.e. the nominator)
_ <i>D32s_Q64s_</i> ←	
KN_NOM_LOW	
Port DIV_N64s⇔	Define for name of of divisor (i.e. the denominator) (signed 32bit)
_ <i>D32s_Q64s_</i> ←	
KN_DIVISOR	
Port DIV_N64s⇔	Define for name of signed 32bit high word of division result
_ <i>D32s_Q64s_</i> ←	
KN_OUT_HIGH	
Port DIV_N64s⇔	Define for name of unsigned 32bit low word of division result
_ <i>D32s_Q64s_</i> ←	
KN_OUT_LOW	
Port DIV_N64s↔	Define for name of integer remainder of the division (unsigned 32bit)
_ <i>D32s_Q64s_</i> ←	
KN_OUT_REM	

```
5.9.2.3 KERNEL_INFO apu_dot_inv_NewtonRaphson ( "apu_dot_inv_NewtonRaphson ", 4, __port(_index(0), __identifier("Inv_Inverse_divisor"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("Inv_divisor"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("Inv_log2fact"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("Inv_shiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1))
```

Inverse computation kernel metadata. Computes inverse of each image pixel with the Newton-Raphson method Outputs signed 32bit inverse of the input in fixed point Q<ShiftFact>.(31-<ShiftFact>) format

COMPUTE_IN↔	Define for Kernel name
V_KN	
4	Number of ports
Port INV_KN_I↔	Define for name of the inverse of div in Q <shiftfact>.(31-<shiftfact>)</shiftfact></shiftfact>
NV_DIV	
Port	Define for name of the image containing 32bit numbers to be inverted
INV_KN_DIV	
Port INV_KN_←	Define for name of the returned image of unsigned 8bit log_2 values of each member of div
LOG2Fact	
Port INV_KN_←	Define for name of the shift factor for the fixed point Q <shiftfact>.(31-<shiftfact>) format</shiftfact></shiftfact>
SHIFTFact	

5.9 Division 47

```
5.9.2.4 KERNEL_INFO apu_dot_log2 ( " apu_dot_log2 " , 2 , __port(__index(0), __identifier("Log2_input"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("Log2_fact"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Computes log2 of the absolute value of each image pixel. Outputs log2(abs(Input)) + 1.

Parameters

LOG2_KN	Define for Kernel name
2	Number of ports
Port	Define for name of input image (signed 32bit).
LOG2_KN_IN	
Port LOG2_KN↔	Define for name of logarithm result (signed 32bit).
_LOG2Fact	

5.9.2.5 void compute64bitLog2 (vec08u * log2Fact, vec32s * input_high, vec32u * input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(abs(input[i])) + 1, where input is a signed 64bit integer

Computes the log_2 + 1 of all the elements of the signed 64bit input. This is a helper function

Parameters

log2Fact	- [Output] the value representing 1+log_2(input)
input_high	- [Input] the high word of the 64bit input vector
input_low	- [Input] the low word of the 64bit input vector
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width(in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width(in elements not bytes) including padding

5.9.2.6 void compute64bitLog2u (vec08u * log2Fact, vec32u * input_high, vec32u * input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(input[i]) + 1, where input is a unsigned 64bit integer

Computes the log_2 + 1 of all the elements of the unsigned 64bit input. This is a helper function

Parameters

log2Fact	- [Output] the value representing 1+log_2(input)
input_high	- [Input] the high word of the 64bit input vector
input_low	- [Input] the low word of the 64bit input vector
bw	- [Input] Block width
bh	- [I nput] Block height
inStrideWidth	- [Input] Source block width(in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width(in elements not bytes) including padding

5.9.2.7 void computeInv_NewtonRaphson (vec32s * invDiv, vec32s * div, vec08u * log2Fact, const int08s shiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise inverse of a signed integer vector using the NewtonRaphson algorithm.

 $\label{lem:compute the inverse of a 32bit integer in Q<ShiftFact>. (31-<ShiftFact>) format with the NewtonRaphson algorithm$

invDiv	- [Output] the inverse of div in Q <shiftfact>.(31-<shiftfact>)</shiftfact></shiftfact>
--------	---

5.9 Division 49

div	- [Input] the vector containing the numbers to be inverted (signed 32bit)
log2Fact	- [Output] Returns the vector of log_2 values of each member of div
shiftFact	- [Input] the shift factor for the fixed point Q <shiftfact>.(31-<shiftfact>) format</shiftfact></shiftfact>
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.9.2.8 void computeLog2 (vec08u * log2Fact, vec32s * input, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(abs(input[i])) + 1, where input is a signed 32bit integer

Pixelwise Computation the log 2 + 1 of all the elements of the input. This is a helper function for computeInv ← NewtonRaphson

Parameters

Parameters

log2Fact	- [Output] the values representing 1+log_2(abs(input))
input	- [Input] the 32bit signed input vector
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width(in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width(in elements not bytes) including padding

5.9.2.9 void computeLog2u (vec08u * log2Fact, vec32u * input, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(input[i]) + 1, where input is a unsigned 32bit integer

Computes the log_2 + 1 of all the elements of the input. This is a helper function for computeInv_NewtonRaphson

log2Fact	- [Output] the values representing 1+log_2(input)
input	- [Input] the 32bit unsigned input vector
bw	- [Input] Block width

[edipat] the values representing trieg_2(input)
- [Input] the 32bit unsigned input vector
- [Input] Block width
- [Input] Block height
- [Input] Source block width(in elements not bytes) including padding
- [Input] Destination block width(in elements not bytes) including padding
_

5.9.2.10 void dot_division_filter (vec32s * res, vec32s * numerator, vec32s * denominator, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise division of signed 32bit integers.

Divide 32 bit numbers with the APEX internal division. quot[i] = numerator[i]/denominator[i], foreach i

auot	- [Output] Destination block pointer
7	
numerator	- [Input] Source block pointer of img A
denominator	- [Input] Source block pointer of img B

bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.9.2.11 void dot_division_filter_N64s_D32s_Q64s (vec32s * dst_high, vec32u * dst_low, vec32u * dst_rem, vec32s * nom_high, vec32u * nom_low, vec32s * divisor, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise division of a 64bit nominator by a 32bit divisor.

Compute the 64bit division of a 64bit integer with a 32bit integer

dst_high	- [Output] High word of destination block pointer
dst_low	- [Output] Low word of destination block pointer
dst_rem	- [Output] Remainder of the division (i.e. decimal part as integer)
nom_high	- [Input] Source block pointer to high word of nominator
nom_low	- [Input] Source block pointer to low word of nominator
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10 Multiplication

5.10.1 Detailed Description

Element-wise multiplication.

Collaboration diagram for Multiplication:



Functions

- KERNEL_INFO apu_dot_mult_in16s_out32s (" apu_dot_mult_in16s_out32s ", 3, __port(_index(0), _
 _ identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 _ type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek
 _ size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_in32s_out32s (" apu_dot_mult_in32s_out32s ", 3, __port(_index(0), _
 __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek
 __size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_in32s_out64s (" apu_dot_mult_in32s_out64s ", 4, __port(__index(0), _ __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __ attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek __size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT_High"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __ identifier("DotMultKn_OUT_Low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_ data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_in32s_in16s_out32s (" apu_dot_mult_in32s_in16s_out32s ", 3, __port(_
 index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), _
 __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), _
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_scalar_in32s_out32s (" apu_dot_mult_scalar_in32s_out32s ", 3, __port(_
 __index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __
 e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultScalKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), _

 $\begin{tabular}{ll} $_ek_size(1,\,1))$, $__port(__index(2), $__identifier("MultScalKn_IN_SCALAR")$, $__attributes(ACF_ATTR_SCL $$$ $_IN_STATIC_FIXED)$, $__spatial_dep(0,\,0,\,0,\,0)$, $__e0_data_type(d32s)$, $__e0_size(1,\,1)$, $__ek_size(1,\,1)$)) $$$

- KERNEL_INFO apu_dot_left_shift_in16u_out16s (" apu_dot_left_shift_in16u_out16s ", 3, __port(__index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 _type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultScalKn_OUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL
 _IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_right_shift_in64s_out64s (" apu_dot_right_shift_in64s_out64s ", 5, __port(__ index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("RightShift_\(\to \) 64bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0\(\to \) _size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("RightShift_64bit_OutHigh"), __attributes(A\(\to \) CF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("RightShift_64bit_OutLOW"), __attributes(ACF_ATTR_VEC_OUT), __\(\to \) spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __\(\to \) identifier("RightShift_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_right_shift_in64s_out32s (" apu_dot_right_shift_in64s_out32s ", 4, __port(__ index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("RightShift_\(\to \) 64bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __\(\to \) e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("RightShift_32bit_Out"), __attributes(ACF\(\to \) ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("RightShift_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FI\(\to \) XED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_out32s (" apu_dot_lsh1_in32s_out32s ", 2, __port(__index(0), _
 __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2Kn_OUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), _
 __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_out64s (" apu_dot_lsh1_in32s_out64s ", 3, __port(__index(0), _
 __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2Kn_Output_high"),
 __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __e
 ek_size(1, 1)), __port(__index(2), __identifier("MultBy2Kn_Output_low"), __attributes(ACF_ATTR_VEC_extrapple OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_Q3_28_out64s (" apu_dot_lsh1_in32s_Q3_28_out64s ", 3, __port(\(\to _\) index(0), __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), _\(\to _\) e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2_Q3_28\(\to _\) Kn_Output_int"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("MultBy2_Q3_28_Kn_Output_frac"), __\(\to _\) attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __\(\to _\) ek_size(1, 1)))
- vbool hasSign (vec32s &a, vec32s &b)

sign(a) * sign(b) == -1

void change64bitSign (vec32s &highWord, vec32u &lowWord)

In place sign change of a 64bit integer, i.e. a = -a.

void dot_mult_in16s_out32s_filter (vec32s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit multiplication => signed 32bit.

void dot_mult_in32s_out32s_filter (vec32s *dst, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 32bit.

void dot_mult_in32s_in16s_out32s_filter (vec32s *dst, vec32s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit and 16bit multiplication => signed 32bit.

void dot_mult_in32s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *srcA, vec32s *srcB, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 64bit.

void dot_mult_in32u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA, vec32u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 32bit multiplication => unsigned 64bit.

void dot_mult_in64s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_←
low, vec32s *srcB high, vec32u *srcB low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 64bit multiplication => signed 64bit.

void dot_mult_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit multiplication => unsigned 64bit.

void dot_mult_scalar_in08u_out16s_filter (vec16s *dst, vec08u *srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 8bit multiplication with a fixed scalar => signed 16bit.

• void dot_mult_scalar_in32s_out32s_filter (vec32s *dst, vec32s *srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication with a fixed scalar => signed 32bit.

• void lsh_in16u_out16s_filter (vec16s *upShifted, vec16u *src, vec16s *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 16bit left shift with a signed 16bit shift vector => signed 16bit.

void lsh_in32u_out32u_filter (vec32u *upShifted, vec32u *src, vec08u *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 32bit.

void lsh_in32s_out32s_filter (vec32s *upShifted, vec32s *src, vec08u *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 32bit.

void lsh_in32s_out64s_filter (vec32s *upShifted_high, vec32u *upShifted_low, vec32s *src, vec08u *left←
 ShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 64bit.

void lsh_in32u_out64u_filter (vec32u *upShifted_high, vec32u *upShifted_low, vec32u *src, vec08u *left←
 ShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 64bit.

• void lsh_in32s_Q3_28_out64s_filter (vec32s *upShifted_int, vec32s *upShifted_frac, vec32s *src, vec08u *leftShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift of a 32bit matrix in Q3_28 format with an unsigned 8bit shift vector => signed 64bit.

void rsh_in32u_out32u_filter (vec32u *downShift, vec32u *src, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit right shift with an unsigned 8bit shift vector => unsigned 32bit.

void rsh_in32s_out32s_filter (vec32s *downShift, vec32s *src, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit right shift with a signed 8bit shift vector => signed 32bit.

• void rsh_in64s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

void rsh_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

void rsh_in64s_out32s_filter (vec32s *dst, vec32s *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 32bit.

5.10.2 Function Documentation

```
5.10.2.1 KERNEL_INFO apu_dot_left_shift_in16u_out16s ( " apu_dot_left_shift_in16u_out16s " , 3 ,
    __port(__index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0),
    __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("MultScalKn_OUT"),
    __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1,
    __port(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED),
    __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Left shift kernel metadata. Shifts to the left each pixel of a unsigned 16bit image by a scalar shift value. Outputs signed 16bit shift result

Warning

No checks are performed for carry-over resulting from the left shift.

Parameters

LEFT_SHIFT_←	Define for Kernel name
In16u_Out16s⇔	
_KN	
3	Number of ports
Port MULT_S↔	Define for name of input image (unsigned 16bit)
CALAR_KN_IN	
Port	Define for name of shift result (signed 16bit)
<i>MULT_SCALA</i> ←	
R_KN_OUT	
Port MULT_S↔	Define for name of scalar used to left shift the image pixels (unsigned 32bit).
CALAR_KN_I↔	
N_SCALAR	

```
5.10.2.2 KERNEL_INFO apu_dot_lsh1_in32s_out32s ( " apu_dot_lsh1_in32s_out32s " , 2 , __port(_index(0), __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("MultBy2Kn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Left shift kernel metadata by one. Shifts to the left each pixel of a signed 32bit image by one. Outputs signed 32bit shift result

Warning

No checks are performed for carry-over resulting from the left shift.

	MULT_BY_2_←	Define for Kernel name
	In32s_Out32s⇔	
	_KN	
Ī	2	Number of ports
ſ	Port MULT_BY←	Define for name of the input image (signed 32bit)
	_2_In32s_ <i>←</i>	
	Out32s_KN_IN	

Port	Define for name of the shift result (signed 32bit)
MULT_BY_2_⇔	
In32s_Out32s⇔	
_KN_OUT	

```
5.10.2.3 KERNEL_INFO apu_dot_lsh1_in32s_out64s ( " apu_dot_lsh1_in32s_out64s " , 3 , __port(__index(0), __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("MultBy2Kn_Output_high"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("MultBy2Kn_Output_low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Left shift kernel metadata by one. Shifts to the left each pixel of a signed 32bit image by one. Outputs signed 64bit shift result

Parameters

MULT_BY_2_↔	Define for Kernel name
In32s_Out64s⇔	
_KN	
3	Number of ports
Port MULT_BY←	Define for name of the input image (signed 32bit)
_2_In32s_ <i>←</i>	
Out32s_KN_IN	
Port	Define name of signed 32bit high word of shift result
MULT_BY_2_←	
In32s_Out32s⇔	
_KN_OUT_H	
Port	Define name of unsigned 32bit low word of shift result
MULT_BY_2_←	
In32s_Out32s⇔	
_KN_OUT_LOW	

```
5.10.2.4 KERNEL_INFO apu_dot_Ish1_in32s_Q3_28_out64s ( " apu_dot_Ish1_in32s_Q3_28_out64s " , 3 , __port(__index(0), __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("MultBy2_Q3_28_Kn_Output_int"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("MultBy2_Q3_28_Kn_Output_frac"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Left shift kernel metadata by one for fixed point numbers in Q3.28 format. Shifts to the left by one each pixel of a signed 32bit image in fixed point Q3.28 format. Outputs signed 64bit shift result in fixed point Q3.28 format

MULT_BY_2_↔	Define for Kernel name
<i>Q3_28_In32s_</i> ↔	
Out64s_KN	
3	Number of ports
Port MULT_BY↔	Define for name of the input image (signed 32bit)
_2_In32s_ <i>←</i>	
Out32s_KN_IN	

Port MULT_BY⇔	Define name of signed 32bit integer part of shift result (in normal integer format)
_2_Q3_28_KN↔	
_OUT_INT	
Port MULT_BY⇔	Define name of unsigned 32bit fractional part of shift result (in fixed point Q3.28 format).
_2_Q3_28_KN↔	
_OUT_FRAC	

```
5.10.2.5 KERNEL_INFO apu_dot_mult_in16s_out32s ( " apu_dot_mult_in16s_out32s " , 3 , __port(_index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Multiplication kernel metadata. Multiplies pixelwise two signed 16bit images. Outputs signed 32bit multiplication result

Parameters

MULT_In16s_←	Define for Kernel name
Out32s_KN	
3	Number of ports
Port	Define for name of first input image name (signed 16bit)
MULT_KN_INA	
Port	Define for name of second input image name (signed 16bit)
MULT_KN_INB	
Port	Define name of multiplication result (signed 32bit).
MULT_KN_OUT	

```
5.10.2.6 KERNEL_INFO apu_dot_mult_in32s_in16s_out32s ( "apu_dot_mult_in32s_in16s_out32s ", 3, __port(__index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Multiplication kernel metadata. Multiplies pixelwise two signed 32bit images. Outputs signed 64bit multiplication result

MULT_In32s_←	Define for Kernel name
Out64s_KN	
4	Number of ports
Port	Define for name of first input image (signed 32bit)
MULT_KN_INA	
Port	Define for name of second input image (signed 32bit)
MULT_KN_INB	
Port MULT_K↔	Define for name of signed 32bit high word of multiplication result
N_OUT_HIGH	
Port MULT_K↔	Define for name of unsigned 32bit low word of multiplication result
N_OUT_LOW	

```
5.10.2.7 KERNEL_INFO apu_dot_mult_in32s_out32s ( " apu_dot_mult_in32s_out32s " , 3 , __port(_index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __e0_size(1, 1), __ek_size(1, 1)) )
```

Multiplication kernel metadata. Multiplies pixelwise two signed 32bit images. Outputs signed 32bit multiplication result

Warning

Does not check out of range values above/below +/-2^(31-1)

Parameters

MULT_In32s_←	Define for Kernel name
Out32s_KN	
3	Number of ports
Port	Define for name of first input image (signed 32bit)
MULT_KN_INA	
Port	Define for name of second input image (signed 32bit)
MULT_KN_INB	
Port	Define for name of multiplication result (signed 32bit).
MULT_KN_OUT	

```
5.10.2.8 KERNEL_INFO apu_dot_mult_in32s_out64s ( " apu_dot_mult_in32s_out64s " , 4 , __port(_index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("DotMultKn_OUT_High"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("DotMultKn_OUT_Low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Multiplication kernel metadata. Multiplies pixelwise two signed 32bit images. Outputs signed 64bit multiplication result

MULT_In32s_←	Define for Kernel name
Out64s_KN	
4	Number of ports
Port	Define for name of first input image (signed 32bit)
MULT_KN_INA	
Port	Define for name of second input image (signed 32bit)
MULT_KN_INB	
Port MULT_K↔	Define for name of signed 32bit high word of multiplication result
N_OUT_HIGH	
Port MULT_K↔	Define for name of unsigned 32bit low word of multiplication result
N_OUT_LOW	

```
5.10.2.9 KERNEL_INFO apu_dot_mult_scalar_in08u_out16s ( " apu_dot_mult_scalar_in08u_out16s " , 3 ,
    __port(__index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0),
    __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("MultScalKn_OUT"),
    __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1,
    __nort(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED),
    __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Scalar multiplication kernel metadata. Multiplies pixelwise a unsigned 8bit image with a scalar value. Outputs signed 16bit multiplication result

Warning

The type of the scalar value is int32s for convenience. It should have values in the maximal range of shorts. No checks are performed for out of range values below/above $\pm -2^{(15-1)}$.

Parameters

<i>MULT_SCALA</i> ↔	Define for Kernel name
R_In08u_⇔	
Out16s_KN	
3	Number of ports
Port MULT_S⇔	Define for name of input image (signed 32bit)
CALAR_KN_IN	
Port	Define for name of multiplication result (signed 32bit)
<i>MULT_SCALA</i> ←	
R_KN_OUT	
Port MULT_S↔	Define for name of scalar used to multiply the image (signed 32bit).
CALAR_KN_I↔	
N_SCALAR	

Scalar multiplication kernel metadata. Multiplies pixelwise a signed 32bit image with a scalar value. Outputs signed 32bit multiplication result

Warning

No checks are performed for out of range values below/above $\pm -2^{(31-1)}$.

<i>MULT_SCALA</i> ←	Define for Kernel name
<i>R_In32s_</i> ↔	
Out32s_KN	
3	Number of ports
Port MULT_S⇔	Define for name of the input image (signed 32bit)
CALAR_KN_IN	

Port	Define for name of multiplication result (signed 32bit)
<i>MULT_SCALA</i> ⇔	
R_KN_OUT	
Port MULT_S↔	Define for name of scalar used to multiply the image (signed 32bit).
CALAR_KN_I↔	
N_SCALAR	

```
5.10.2.11 KERNEL_INFO apu_dot_right_shift_in64s_out32s ( " apu_dot_right_shift_in64s_out32s " , 4 , __port(_index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("RightShift_64bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("RightShift_32bit_Out"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("RightShift_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Right shift kernel metadata. Shifts to the right each pixel of a signed 64bit image by a scalar shift value. Outputs signed 32bit shift result

Parameters

RIGHT_SHIFT⇔	Define for Kernel name
_ <i>In64s</i> _←	
Out32s_KN	
5	Number of ports
Port	Define for name of the signed 32bit high word of the input image
RIGHT_SHIFT⇔	
_In64s_HIGH	
Port	Define for name of the unsigned 32bit low word of the input image
RIGHT_SHIFT⇔	
_In64s_LOW	
Port	Define for name of the signed 32bit high word of the shift result
RIGHT_SHIFT⇔	
_Out64s_HIGH	
Port	Define for name of the unsigned 32bit low word of the shift result
RIGHT_SHIFT⇔	
_Out64s_LOW	
Port RIGHT_S⇔	Define for name of the unsigned 32bit scalar used to left shift the image pixels.
HIFT_Fact	

```
5.10.2.12 KERNEL_INFO apu_dot_right_shift_in64s_out64s ( "apu_dot_right_shift_in64s_out64s ", 5, __port(_index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("RightShift_64bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("RightShift_64bit_OutHigh"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("RightShift_64bit_OutLOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("RightShift_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1))
```

Right shift kernel metadata. Shifts to the right each pixel of a signed 64bit image by a scalar shift value. Outputs signed 64bit shift result

Parameters

RIGHT_SHIFT↔	Define for Kernel name
_ <i>In64s</i> _←	
Out64s_KN	
5	Number of ports
Port	Define for name of the signed 32bit high word of the input image
RIGHT_SHIFT⇔	
_In64s_HIGH	
Port	Define for name of the unsigned 32bit low word of the input image
RIGHT_SHIFT⇔	
_In64s_LOW	
Port	Define for name of the signed 32bit high word of the shift result
RIGHT_SHIFT⇔	
_Out64s_HIGH	
Port	Define for name of the unsigned 32bit low word of the shift result
RIGHT_SHIFT⇔	
_Out64s_LOW	
Port RIGHT_S⇔	Define name of the unsigned 32bit scalar used to left shift the image pixels.
HIFT_Fact	

5.10.2.13 void change64bitSign (vec32s & highWord, vec32u & lowWord)

In place sign change of a 64bit integer, i.e. a = -a.

Change the sign of 64bit values stored in two vectors, a high and a low-word vector.

Parameters

ſ	highWord	- [Input/output] high word input/output. Result is stored directly into same vector
	lowhWord	- [Input/output] low word input/output. Result is stored directly into same vector

5.10.2.14 void dot_mult_in16s_out32s_filter (vec32s * dst, vec16s * srcA, vec16s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit multiplication => signed 32bit.

Dot multiplication between two 16bit matrices (i.e multiply elementwise the matrix coeffs), with signed 32bit result. dst[i] = srcImageA[i] * srcImageB[i]

Parameters

dst	- [Output] signed 32bit destination block pointer
srcA	- [Input] signed 16bit source block pointer of img A
srcB	- [Input] signed 16bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.15 void dot_mult_in32s_in16s_out32s_filter (vec32s * dst, vec32s * srcA, vec16s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit and 16bit multiplication => signed 32bit.

Dot multiplication between a signed 32bit and a signed 16bit matrix (i.e multiply elementwise the matrix coeffs), with signed 32bit result

Warning

No out of range is taken into consideration!

Parameters

dst	- [Output] signed 32bit destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
srcB	- [Input] signed 16bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.16 void dot_mult_in32s_out32s_filter (vec32s * dst, vec32s * srcA, vec32s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 32bit.

Dot multiplication between two 32bit matrices (i.e multiply elementwise the matrix coeffs), with signed 32bit result

Warning

No out of range is taken into consideration!

Parameters

dst	- [Output] signed 32bit destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
srcB	- [Input] signed 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.17 void dot_mult_in32s_out64s_filter (vec32s * dst_high, vec32u * dst_low, vec32s * srcA, vec32s * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 64bit.

Dot 64bit multiplication between two signed 32bit matrices (i.e multiply elementwise the matrix coeffs). Results have 64bits and are organized into two 32bit matrices (i.e the lower and the higher word matrices of the multiplication result)

dst_high	- [Output] signed 32bit high word of destination block pointer
dst_low	- [Output] signed 32bit low word of destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
srcB	- [Input] signed 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding

outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.18 void dot_mult_in32u_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32u * srcA, vec32u * srcA, vec32u * srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 32bit multiplication => unsigned 64bit.

Dot 64bit multiplication between two unsigned 32bit matrices (i.e multiply elementwise the matrix coeffs). Results have 64bits and are organized into two 32bit matrices (i.e the lower and the higher word matrices of the multiplication result)

Parameters

dst_high	- [Output] unsigned 32bit high word of destination block pointer
dst_low	- [Output] signed 32bit low word of destination block pointer
srcA	- [Input] unsigned 32bit source block pointer of img A
srcB	- [Input] unsigned 32bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.19 void dot_mult_in64s_out64s_filter (vec32s * dst_high, vec32u * dst_low, vec32s * srcA_high, vec32u * srcA_low, vec32s * srcB_high, vec32u * srcB_low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 64bit multiplication => signed 64bit.

Dot 64bit multiplication between two signed 64bit matrices (i.e multiply elementwise the matrix coeffs). Results have 64bits and are organized into two 32bit matrices (i.e the lower and the higher word matrices of the multiplication result)

Warning

out of ranges over the 64bit limit are not taken into account!

Parameters

dst high	- [Output] signed 32bit high word of destination block pointer
dst_low	- [Output] unsigned 32bit low word of destination block pointer
srcA_high	- [Input] signed 32bit high word of 64bit source block pointer of img A
srcA_low	- [Input] unsigned 32bit low word of 64bit source block pointer of img A
srcB_high	- [Input] signed 32bit high word of 64bit source block pointer of img B
srcB_low	- [Input] unsigned 32bit low word of 64bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.20 void dot_mult_in64u_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32u * srcA_high, vec32u * srcA_high, vec32u * srcA_high, vec32u * srcB_low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit multiplication => unsigned 64bit.

Dot 64bit multiplication between two unsigned 64bit matrices (i.e multiply elementwise the matrix coeffs). Results have 64bits and are organized into two 32bit matrices (i.e the lower and the higher word matrices of the multiplication result)

Warning

out of ranges over the 64bit limit are not taken into account!

Parameters

dst_high	- [Output] unsigned 32bit high word of 64bit destination block pointer
dst_low	- [Output] unsigned 32bit low word of 64bit destination block pointer
srcA_high	- [Input] unsigned 32bit high word of 64bit source block pointer of img A
srcA_low	- [Input] unsigned 32bit low word of 64bit source block pointer of img A
srcB_high	- [Input] unsigned 32bit high word of 64bit source block pointer of img B
srcB_low	- [Input] unsigned 32bit low word of 64bit source block pointer of img B
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.21 void dot_mult_scalar_in08u_out16s_filter (vec16s * dst, vec08u * srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 8bit multiplication with a fixed scalar => signed 16bit.

Dot multiplication of all elements of a 8bit matrix with a scalar value, with signed 16bit output. dst[i] = srcImage0[i] * scalar;

Warning

The scalar value should have ideally a 8bit value, or maximally a 16bit value. No out of range is checked!

Parameters

dst	- [Output] signed 16bit destination block pointer
srcA	- [Input] unsigned 8bit source block pointer of img A
scalar	- [Input] signed 32bit the scalar value to be multiplied with the matrix
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.22 void dot_mult_scalar_in32s_out32s_filter (vec32s * dst, vec32s * srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication with a fixed scalar => signed 32bit.

Dot multiplication of all elements of a 32bit matrix with a 32bit scalar value. Output is of signed 32bits

Warning

No out of range is checked above the 32bit value range!

Parameters

Generated on Fri Dec 14 2018 18:20:12 for ACF/APEX Kernel SDK by Doxygen

dst	- [Output] 16bit destination block pointer
srcA	- [Input] 8bit source block pointer of img A
scalar	- [Input] 32bit the scalar value to be multiplied with the matrix
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.23 vbool hasSign (vec32s & a, vec32s & b)

sign(a) * sign(b) == -1

Tests if a times/div b would give a negative result

Parameters

а	- [Input] first operand
b	- [Input] second operand

Returns

a boolean vector where the elements reflect the test operation result

5.10.2.24 void lsh_in16u_out16s_filter (vec16s * upShifted, vec16u * src, vec16s * leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 16bit left shift with a signed 16bit shift vector => signed 16bit.

Dot left-shift unsigned 16bit operator by a signed 16bit input vector (each element of one chunk_i is shifted by the shift_vect(i) factor) with signed 16bit output. dst_chunk[i] = srcImage0_chunk[i] << leftShiftFact[i]

Warning

No out of range is checked above the 16bit value range!

Parameters

upShifted	- [Output] signed 16bit destination block pointer for the upShifteded result
src	- [Input] unsigned 16bit source block pointer
leftShiftFact	- [Input] signed 16bit vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.25 void lsh_in32s_out32s_filter (vec32s * upShifted, vec32s * src, vec08u * leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 32bit.

Dot left-shift signed 32bit operator (each element of one chunk_i is shifted by the shift_vect(i) factor) with signed 32bit output

Warning

No out of range is checked above the 32bit value range!

Parameters

unChifted [Output] signed 20hit destination block pointer for the unChifteded regult	
upShifted - [Output] signed 32bit destination block pointer for the upShifteded result	
src - [Input] signed 32bit source block pointer	
leftShiftFact - [Input] unsigned 8bit vector of shift values, one for each chunk	
bw - [Input] Block width	
bh - [Input] Block height	
inStrideWidth - [Input] Source block width (in elements not bytes) including padding	
outStrideWidth - [Input] Destination block width (in elements not bytes) including padding	

5.10.2.26 void lsh_in32s_out64s_filter (vec32s * upShifted_high, vec32u * upShifted_low, vec32s * src, vec08u * leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 64bit.

Dot left-shift signed 32bit operator (each element of one chunk_i is shifted by the shift_vect(i) factor) with signed 64bit output

Parameters

upShifted_high	- [Output] signed 32bit high word of destination block pointer for the upShifted result
upShifted_low	- [Output] unsigned 32bit low word of destination block pointer for the upShifted result
src	- [Input] signed 32bit source block pointer
leftShiftFact	- [Input] unsigned 8bit vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.27 void lsh_in32s_Q3_28_out64s_filter (vec32s * upShifted_int, vec32s * upShifted_frac, vec32s * src, vec08u * leftShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift of a 32bit matrix in Q3_28 format with an unsigned 8bit shift vector => signed 64bit.

Dot left-shift operator for a Q3_28 fixed point input parameter format with 64bit output The output "high" word contains [x] - integer part of the left shifted fixed point input number and the output "low" word contains $\{x\}$ - fractional part of the left shifted fixed point input number

Parameters

upShifted_int	- [Output] [x] - signed 32bit integer part of the left shifted fixed point input number
upShifted_frac	- [Output] {x} - signed 32bit fractional part of the left shifted fixed point input number
src	- [Input] signed 32bit source block pointer containing fixed point numbers in Q3_28 format
leftShiftFact	- [Input] unsigned 8bit vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.28 void lsh_in32u_out32u_filter (vec32u * upShifted, vec32u * src, vec08u * leftShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 32bit.

Dot left-shift unsigned 32bit operator (each element of one chunk_i is shifted by the shift_vect(i) factor) with unsigned 32bit output

Warning

No out of range is checked above the 32bit value range!

Parameters

upShifted	- [Output] unsigned 32bit destination block pointer for the upShifteded result
src	- [Input] unsigned 32bit source block pointer
leftShiftFact	- [Input] unsigned 8bit vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.29 void lsh_in32u_out64u_filter (vec32u * upShifted_high, vec32u * upShifted_low, vec32u * src, vec08u * leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 64bit.

Dot left-shift unsigned 32bit operator (each element of one chunk_i is shifted by the shift_vect(i) factor) with unsigned 64bit output

Parameters

upShifted	- [Output] unsigned 32bit destination block pointer for the upShifted result
src	- [Input] unsigned 32bit source block pointer
leftShiftFact	- [Input] the vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.30 void rsh_in32s_out32s_filter (vec32s * downShift, vec32s * src, vec08u * rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit right shift with a signed 8bit shift vector => signed 32bit.

Dot right-shift signed 32bit operators (each element of one chunk_i is shifted by the shift_vect(i) factor) with signed 32bit output

Parameters

downShift	- [Output] 32bit destination block pointer for the downshifted result
src	- [Input] 32bit source block pointer
rightShiftFact	- [Input] the vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.31 void rsh_in32u_out32u_filter (vec32u * downShift, vec32u * src, vec08u * rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit right shift with an unsigned 8bit shift vector => unsigned 32bit.

Dot right-shift unsigned 32bit operators (each element of one chunk_i is shifted by the shift_vect(i) factor) with unsigned 32bit output dst_chunk[i] = srclmage0_chunk[i] >> rightShiftFact[i]

Parameters

downShift	- [Output] 32bit destination block pointer for the downshifted result
src	- [Input] 32bit source block pointer
rightShiftFact	-[Input] the vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.32 void rsh_in64s_out32s_filter (vec32s * dst, vec32s * in_high, vec32u * in_low, vec08u * rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 32bit.

Dot right-shift operator for 64bit signed input (each element of one chunk_i is shifted by the shift_vect(i) factor)

Warning

No check is performed to see that the right shifted values became less than $2^{^{\wedge}}32$

Parameters

dst	- [Output] signed 32bit destination block pointer for the downshifted result
in_high	- [Input] signed 32bit high word of source block pointer
in_low	- [Input] unsigned 32bit low word of source block pointer
rightShiftFact	- [Input] unsigned 8bit shift values, one for each block
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.33 void rsh_in64s_out64s_filter (vec32s * dst_high, vec32u * dst_low, vec32s * in_high, vec32u * in_low, vec08u * rightShiftFact, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

Dot right-shift operator for 64bit signed input (each element of one chunk_i is shifted by the shift_vect(i) factor)

Parameters

dst_high	- [Output] 32bit signed high word of destination block pointer for the downshifted result
dst_low	- [Output] 32bit unsigned low word of destination block pointer for the downshifted result
in_high	- [Input] 32bit signed high word of source block pointer
in_low	- [Input] 32bit unsigned low word of source block pointer
rightShiftFact	- [Input] the vector of shift values, one for each chunk
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.10.2.34 void rsh_in64u_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32u * in_high, vec32u * in_low, vec32u * in_

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

Dot right-shift operator for 64bit unsigned input (each element of one chunk_i is shifted by the shift_vect(i) factor)

dst_high	- [Output] unsigned 32bit high word of destination block pointer for the downshifted result
dst_low	- [Output] unsigned 32bit low word of destination block pointer for the downshifted result
in_high	- [Input] unsigned 32bit high word of source block pointer
in_low	- [Input] unsigned 32bit low word of source block pointer
rightShiftFact	- [Input] unsigned 8bit shift values, one for each block
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block width (in elements not bytes) including padding
outStrideWidth	- [Input] Destination block width (in elements not bytes) including padding

5.11 Square 69

5.11 Square

5.11.1 Detailed Description

Element-wise square.

Collaboration diagram for Square:



Functions

- KERNEL_INFO apu_dot_sqr_in32s_out64u (" apu_dot_sqr_in32s_out64u ", 3, __port(__index(0), __
 identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s),
 __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_High"), __attributes(ACF_AT
 TR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(2), __identifier("OUTPUT_Low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- vec32u vsqrt 32 (vec32u a)
- void dot_sqr_in16s_out32u_filter (vec32u *dst, vec16s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit square => unsigned 32bit.

void dot_sqr_in32s_out32u_filter (vec32u *dst, vec32s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 32bit.

void dot_sqr_in32s_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 64bit.

void dot_sqr_in32u_out64u_filter (vec32u *out_high, vec32u *out_low, vec32u *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 32bit square => unsigned 64bit.

void dot_sqr_in64s_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_←
low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 64bit square => unsigned 64bit.

void dot_sqr_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_←
 low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit square => unsigned 64bit.

5.11.2 Function Documentation

```
5.11.2.1 KERNEL_INFO apu_dot_sqr_in16s_out32u ( " apu_dot_sqr_in16s_out32u " , 2 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Square kernel metadata. Computespixelwise the square of a input signed 16bit images. Outputs unsigned 32bit square result

5.11 Square 71

Parameters

SQR_In16s_←	Define for Kernel name
Out32u_KN	
2	Number of ports
Port	Define for name of input image (signed 16bit)
SQR_KN_IN	
Port	Define for name of square of input result (unsigned 32bit)
SQR_KN_OUT	

```
5.11.2.2 KERNEL_INFO apu_dot_sqr_in32s_out32u ( " apu_dot_sqr_in32s_out32u " , 2 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Square kernel metadata. Computespixelwise the square of a input signed 32bit images. Outputs unsigned 32bit square result

Warning

Does not check out of range values above/below +/-2^(31-1)

Parameters

SQR_In32s_←	Define for Kernel name
Out32u_KN	
2	Number of ports
Port	Define for name of input image (signed 32bit)
SQR_KN_IN	
Port	Define for name of square of input result (unsigned 32bit)
SQR_KN_OUT	

```
5.11.2.3 KERNEL_INFO apu_dot_sqr_in32s_out64u ( "apu_dot_sqr_in32s_out64u " , 3 , __port(_index(0), __identifier("INPUT"), _attributes(ACF_ATTR_VEC_IN), _spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), _identifier("OUTPUT_High"), _attributes(ACF_ATTR_VEC_OUT), _spatial_dep(0, 0, 0, 0), _e0_data_type(d32u), _e0_size(1, 1), _ek_size(1, 1)) , __port(_index(2), _identifier("OUTPUT_Low"), _attributes(ACF_ATTR_VEC_OUT), _spatial_dep(0, 0, 0, 0), _e0_data_type(d32u), _e0_size(1, 1), _ek_size(1, 1)) )
```

Square kernel metadata. Computespixelwise the square of a input signed 32bit images. Outputs unsigned 64bit square result

SQR_In32s_←	Define for Kernel name
Out32u_KN	
2	Number of ports
Port	Define for name of input image (signed 32bit)
SQR_KN_IN	
Port SQR_KN⇔	Define for name of high word of square of input result (unsigned 32bit)
_OUT_HIGH	
Port SQR_KN↔	Define for name of low word of square of input result (unsigned 32bit)
_OUT_LOW	

5.11.2.4 void dot_sqr_in16s_out32u_filter (vec32u * dst, vec16s * srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit square => unsigned 32bit.

Computes the 32bit elementwise-square of the 16bit input matrix. dst[i] = srcImageA[i].^2;

Warning

Out of range values are not taken into account!

Parameters

dst	- [Output] 32bit Destination block pointer
srcA	- [Input] 32bit Source block pointer of img A
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.5 void dot_sqr_in32s_out32u_filter (vec32u * dst, vec32s * srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 32bit.

Computes the 32bit elementwise-square of the signed 16bit input matrix.

Warning

Out of range values are not taken into account!

Parameters

dst	- [Output] 32bit Destination block pointer
srcA	- [Input] 32bit Source block pointer of img A
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.6 void dot_sqr_in32s_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32s * srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 64bit.

Computes the 64 bit elementwise-square of the signed 32bit input matrix.

dst_high	- [Output] unsigned 32bit high word destination block pointer
dst_low	- [Output] unsigned 32bit low word destination block pointer
srcA	- [Input] signed 32bit source block pointer of img A
bw	- [Input] Block width

5.11 Square 73

bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.7 void dot_sqr_in32u_out64u_filter (vec32u * out_high, vec32u * out_low, vec32u * srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 32bit square => unsigned 64bit.

Computes the 64 bit elementwise-square of the unsigned 32bit input matrix

Parameters

dst_high	- [Output] unsigned 32bit high-word destination block pointer
dst_low	- [Output] unsigned 32bit low-word destination block pointer
srcA	- [Input] unsigned 32bit source block pointer of img A
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.8 void dot_sqr_in64s_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32s * srcA_high, vec32u * srcA_low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 64bit square => unsigned 64bit.

Computes the 64 bit elementwise-square of the signed 64bit input matrix

Warning

Out of range values are not taken into account!

Parameters

dst_high	- [Output] unsigned 32bit high-word destination block pointer
dst_low	- [Output] unsigned 32bit low-word destination block pointer
srcA_high	- [Input] Block pointer to sighed 32bit high word of source
srcA_low	- [Input] Block pointer to unsigned 32bit low word of source
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.9 void dot_sqr_in64u_out64u_filter (vec32u * dst_high, vec32u * dst_low, vec32u * srcA_high, vec32u * srcA_high, vec32u * srcA_high, vec32u * srcA_low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit square => unsigned 64bit.

Computes the 64 bit elementwise-square of the unsigned 64bit input matrix

Warning

Out of range values are not taken into account!

Parameters

dst_high	- [Output] unsigned 32bit high-word destination block pointer
dst_low	- [Output] unsigned 32bit low-word destination block pointer
srcA_high	- [Input] Block pointer to unsigned 32bit high word of source
srcA_low	- [Input] Block pointer to unsigned 32bit low word of source
bw	- [Input] Block width
bh	- [Input] Block height
inStrideWidth	- [Input] Source block stride width (i.e. nr. of elements including padding)
outStrideWidth	- [Input] Destination block stride width (i.e. nr. of elements including padding)

5.11.2.10 vec32u vsqrt_32 (vec32u a)

Computes the 32bit integer square root of the input parameter

va Source vector

Returns

sqrt(va)

5.12 Maximum 75

5.12 Maximum

5.12.1 Detailed Description

Element-wise maximum.

Collaboration diagram for Maximum:



Functions

- KERNEL_INFO apu_max (" apu_max ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_
 ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __

 port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __

 e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_0"), __

 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __

 ek_size(1, 1)))
- void max (vec08u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)
 Element-wise maximum.

5.12.2 Function Documentation

```
5.12.2.1 KERNEL_INFO apu_max ( "apu_max ", 3, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1))))
```

Element-wise maximum kernel metadata.

Parameters

apu_max	Define for Kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
MAX_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
MAX_KN_INB	
Port	Define for name of output image (unsigned 8bit)
MAX_KN_OUT	

5.12.2.2 void max (vec08u * dst, vec08u * srcImage0, vec08u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Element-wise maximum.

Element-wise maximum. out[i] = max(in0[i], in1[i])

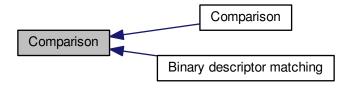
dst	- [Output] Pointer to the destination buffer
srcImage0	- [Input] Pointer to the first source buffer
srcImage1	- [Input] Pointer to the second source buffer
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block
inStrideW	- [Input] Line stride of the source data
outStrideW	- [Input] Line stride of the destination data

5.13 Comparison 77

5.13 Comparison

5.13.1 Detailed Description

Collaboration diagram for Comparison:



Modules

Comparison

Element-wise comparison.

· Binary descriptor matching

Binary descriptor matching.

5.14 Comparison

5.14.1 Detailed Description

Element-wise comparison.

Collaboration diagram for Comparison:



Functions

- KERNEL_INFO apu_lower (" apu_lower ", 3, __port(_index(0), __identifier("LOWER_KN_IN_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek
 __size(1, 1)), __port(_index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_IN), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __
 identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_
 data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_scalar (" apu_lower_scalar ", 3, __port(__index(0), __identifier("LOWER_KN_I ← N_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_Scalar"), __attributes(ACF_ATTR_SCL_IN ← STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __ ← port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_in16s (" apu_lower_in16s ", 3, __port(_index(0), __identifier("LOWER_KN_ ← IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_ ← IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0 ← data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_in32s (" apu_lower_in32s ", 3, __port(__index(0), __identifier("LOWER_KN_ ← IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_ ← IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0 ← data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_abs_lower_in32s ("apu_abs_lower_in32s ", 3, __port(__index(0), __identifier("LOWE← R_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATT← R_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

5.14 Comparison 79

KERNEL_INFO apu_lowerEqual_in32s (" apu_lowerEqual_in32s ", 3, __port(_index(0), __identifier("LO ← WER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __ ← e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_AT ← TR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ← __index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

- KERNEL_INFO apu_mask8b (" apu_mask8b ", 3, __port(_index(0), __identifier("MASK_IN_IMG"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(_index(1), __identifier("MASK_IN_MASK"), __attributes(ACF_ATTR_VEC_IN), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), \leftarrow
 identifier("MASK_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_\leftarrow
 type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_and (" apu_and ", 3, __port(__index(0), __identifier("AND_IN_0"), __attributes(ACF \(\text{ATTR_VEC_IN} \), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), \(\text{port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("AND_OUT"), __\(\text{attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __\(\text{ek size(1, 1)} \) ek_size(1, 1)))
- KERNEL_INFO apu_and_in16u_out16u (" apu_and_in16u_out16u ", 3, __port(_index(0), __identifier("A← ND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0← size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VE← C_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0← data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_and_3Pt_in16u_out16u (" apu_and_3Pt_in16u_out16u ", 4, __port(__index(0), __
 identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u),
 _e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_
 _VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("AND_IN_2"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("AND_OUT"), __ex_attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ex_size(1, 1)))
- KERNEL_INFO apu_and_in08u_out16u (" apu_and_in08u_out16u ", 3, __port(_index(0), __identifier("A⇔ ND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0⇔ __size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VE⇔ C_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__⇔ index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0⇔ __data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_and_in08u_in16u_out16u ("apu_and_in08u_in16u_out16u", 3, __port(_index(0), __
 identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_\top VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__\top index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_\top data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_in64s (" apu_lower_in64s ", 5, __port(__index(0), __identifier("LOWER_KN_I ← N_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0 ← __size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_0_LOW"), __attributes(A ← CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("LOWER_KN_IN_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __ ← spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __ ← identifier("LOWER_KN_IN_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_ ← data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1))
- KERNEL_INFO apu_lower_in64u (" apu_lower_in64u ", 5, __port(__index(0), __identifier("LOWER_KN ← _ IN_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __ ← e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_0_LOW"), __attributes(A ← CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1), __ek_size(1, 1)

```
1)), __port(__index(2), __identifier("LOWER_KN_IN_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __ \hookleftarrow spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __ \hookleftarrow identifier("LOWER_KN_IN_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_ \hookleftarrow data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
```

• void lower (vbool *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "<" operation => bool.

void lower_scalar (vec08u *dst, vec08u *srcImage, unsigned char scalar, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "<" operation => bool.

void lower_in16s (vbool *dst, vec16s *srcImage0, vec16s *srcImage1, int bw, int bh, int inStrideW, int out
 StrideW)

Elementwise signed 16bit "<" operation => bool.

void lower_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int out
 StrideW)

Elementwise signed 32bit "<" operation => bool.

void absLower_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between the absolute values of the operands => bool.

void absLower_in32s_scalar16u (vbool *dst, vec32s *srcImage0, int16u compVal, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between an image and a fixed unsigned 16bit scalar => bool.

void lowerEqual_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<=" operation => bool.

• void lower_in64u (vbool *dst, vec32u *srcImage0_high, vec32u *srcImage0_low, vec32u *srcImage1_high, vec32u *srcImage1 low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 64bit "<" operation => bool.

• void lower_in64s (vbool *dst, vec32s *srcImage0_high, vec32u *srcImage0_low, vec32s *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit "<" operation => bool.

- void mask_kn (vec08u *dst, vec08u *srcImage, vec08u *srcMask, int bw, int bh, int inStrideW, int outStrideW)

 Elementwise unsigned 8bit mask operation => unsigned 8bit.

Elementwise unsigned 8bit "&&" operation => unsigned 8bit.

void and_in16u_out16u (vec16u *dst, vec16u *srcImage0, vec16u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 16bit "&&" operation => unsigned 16bit.

void and_in08u_out16u (vec16u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "&&" operation => unsigned 16bit.

Elementwise unsigned 8bit "&&" unsigned 16bit operation => unsigned 16bit.

• void and_3Pt_in16u_out16u (vec16u *dst, vec16u *srcImage0, vec16u *srcImage1, vec16u *srcImage2, int bw, int inStrideW, int outStrideW)

Elementwise unsigned 3-point 16bit "&&" operation => unsigned 16bit.

5.14 Comparison 81

5.14.2 Function Documentation

5.14.2.1 void absLower_in32s (vbool * dst, vec32s * srcImage0, vec32s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between the absolute values of the operands => bool.

Comparison btw the absolute values of two 32bit matrices: dst[i] = (abs(srcImage0[i]) < abs(srcImage1[i]))

Parameters

dst	- [Output] boolean destination block pointer
srcImage0	- [Input] signed 32bit source block pointer of img 0
srcImage1	- [Input] signed 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.2 void absLower_in32s_scalar16u (vbool * dst, vec32s * srcImage0, int16u compVal, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between an image and a fixed unsigned 16bit scalar => bool.

Comparison btw the absolute values of a signed 32bit matrix and a scalar unsigned 16bit value: dst[i] = (abs(src⇔ Image0[i]) < val)

Parameters

dst	- [Output] boolean destination block pointer
srcImage0	- [Input] signed 32bit source block pointer of img 0
compVal	- [Input] unsigned 16bit value to compare to
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.3 void and_3Pt_in16u_out16u (vec16u * srclmage0, vec16u * srclmage1, vec16u * srclmage2, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 3-point 16bit "&&" operation => unsigned 16bit.

Elementwise AND btw three unsigned 16bit numbers: dst[i] = (srcImage0[i]!= 0 && srcImage1[i]!= 0 && srcImage2[i]!= 0). Result is converted to unsigned 16bit

dst	- [Output] 16bit Destination block pointer
srcImage0	- [Input] 16bit source block pointer of first image
srcImage1	- [Input] 16bit source block pointer of second image
srcImage2	- [Input] 16bit source block pointer of third image
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding

outStrideW	- [Input] Destination block width (in elements not bytes) including padding
ouiSiriaevv	- Imput Destination block width (in elements not bytes) including paddi

5.14.2.4 void and_in08u_in16u_out16u (vec16u * dst, vec08u * srcImage0, vec16u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "&&" unsigned 16bit operation => unsigned 16bit.

Elementwise AND btw one unsigned 8bit and one unsigned 16bit blocks: dst[i] = (srcImage0[i] && srcImage1[i]). Result is converted to unsigned 16bit

Parameters

dst	- [Output] unsigned 16bit destination block pointer
srcImage0	- [Input] unsigned 8bit source block pointer of first image
srcImage1	- [Input] unsigned 16bit source block pointer of second image
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.5 void and_in08u_out16u (vec16u * dst, vec08u * srcImage0, vec08u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "&&" operation => unsigned 16bit.

Elementwise AND btw two unsigned 16bit numbers: dst[i] = (srcImage0[i] && srcImage1[i]). Result is converted to unsigned 16bit

Parameters

dst	- [Output] unsigned 16bit destination block pointer
srcImage0	- [Input] signed 8bit source block pointer of first image
srcImage1	- [Input] signed 8bit source block pointer of second image
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.6 void and_in16u_out16u (vec16u * dst, vec16u * srcImage0, vec16u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 16bit "&&" operation => unsigned 16bit.

Elementwise AND btw two unsigned 16bit numbers: dst[i] = (srcImage0[i] && srcImage1[i])

dst	- [Output] unsigned 16bit destination block pointer
srcImage0	- [Input] unsigned 16bit source block pointer of first image
srcImage1	- [Input] unsigned 16bit source block pointer of second image
bw	- [Input] Block width
bh	- [Input] Block height

5.14 Comparison 83

instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.7 void and_kn (vec08u * srclmage0, vec08u * srclmage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "&&" operation => unsigned 8bit.

Elementwise AND btw two unsigned 8bit numbers: dst[i] = (srcImage0[i] && srcImage1[i])

Parameters

dst	- [Output] unsigned 8bit destination block pointer
srcImage0	- [Input] unsigned 8bit source block pointer of first image
srcImage1	- [Input] unsigned 8bit source block pointer of second image
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

```
5.14.2.8 KERNEL_INFO apu_abs_lower_in32s ( " apu_abs_lower_in32s " , 3 , __port(_index(0), __identifier("LOWER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Absolute Lower kernel metadata. Compares pixelwise the absolute values of two signed 32bit images. Outputs unsigned 8bit comparison result. Is true if abs(INPUTA) <= abs(INPUTB)

Parameters

ABS_LOWER↔	Define for Kernel name
_In32s_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (signed 32bit)
KN_INA	
Port LOWER_←	Define for name of second input image (signed 32bit)
KN_INB	
Port LOWER_←	Define for name ofcomparison result of the two images (unsigned 8bit).
KN_OUT	

Absolute Lower kernel metadata. Compares pixelwise the absolute values of signed 32bit image with a single scalar unsigned 16bit value. Outputs unsigned 8bit comparison result. Is true if abs(INPUTA) <= scalar

Parameters

Generated on Fri Dec 14 2018 18:20:12 for ACF/APEX Kernel SDK by Doxygen

ABS_LOWER↔	Define for Kernel name
_ <i>In32s</i> _⇔	
scalar16u_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (signed 32bit)
KN_INA	
Port LOWER_←	Define for name of scalar value (unsigned 16bit)
KN_INB	
Port LOWER_←	Define for name of comparison result of the two images (unsigned 8bit).
KN_OUT	

```
5.14.2.10 KERNEL_INFO apu_and ( " apu_and " , 3 , __port(__index(0), __identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

AND operator kernel metadata. Pixelwise "AND" operator between two unsigned 8bit images. Outputs unsigned 8bit comparison result. Is true if (INPUTA != 0) && (INPUTB != 0)

Parameters

AND_K	Define for Kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 16bit)
AND_KN_INA	
Port	Define for name of second input image (unsigned 16bit)
AND_KN_INB	
Port	Define for name of comparison result of the two images (unsigned 16bit).
AND_KN_OUT	

```
5.14.2.11 KERNEL_INFO apu_and_3Pt_in16u_out16u ( "apu_and_3Pt_in16u_out16u ", 4, __port(_index(0), __identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("AND_IN_2"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

AND operator kernel metadata. Pixelwise "AND" operator between three unsigned 16bit images. Outputs unsigned 16bit comparison result. Is true if (INPUTA != 0) && (INPUTB != 0) && (INPUTC != 0)

AND_3Pt_←	Define for Kernel name
In16u_Out16u⇔	
_K	
3	Number of ports
Port	Define for name of first input image (unsigned 16bit)
AND_KN_INA	
Port	Define for name of second input image (unsigned 16bit)
AND_KN_INB	

5.14 Comparison 85

Port	Define for name of third input image (unsigned 16bit)
AND_KN_INC	
Port	Define for name of comparison result of the two images (unsigned 16bit).
AND_KN_OUT	

```
5.14.2.12 KERNEL_INFO apu_and_in08u_in16u_out16u ( "apu_and_in08u_in16u_out16u ", 3, __port(_index(0), __identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

AND operator kernel metadata. Pixelwise "AND" operator between two unsigned 16bit images. Outputs unsigned 16bit and operator result. Is true if (INPUTA != 0) && (INPUTB != 0)

Parameters

AND_In08u_⇔	Define for Kernel name
In16u_Out16u⇔	
_KN	
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
AND_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
AND_KN_INB	
Port	Define for name of comparison result of the two images (unsigned 16bit).
AND_KN_OUT	

```
5.14.2.13 KERNEL_INFO apu_and_in08u_out16u ( " apu_and_in08u_out16u " , 3 , __port(__index(0), __identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

AND operator kernel metadata. Pixelwise "AND" operator between two unsigned 8bit images. Outputs unsigned 16bit and operator result. Is true if (INPUTA != 0) && (INPUTB != 0)

AND_In08u_←	Define for Kernel name
Out16u_KN	
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
AND_KN_INA	
Port	Define for name of second input image (unsigned 8bit)
AND_KN_INB	
Port	Define for name of comparison result of the two images (unsigned 16bit).
AND_KN_OUT	

```
5.14.2.14 KERNEL_INFO apu_and_in16u_out16u ( "apu_and_in16u_out16u ", 3, __port(__index(0), __identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

AND operator kernel metadata. Pixelwise "AND" operator between two unsigned 16bit images. Outputs unsigned 16bit comparison result. Is true if (INPUTA != 0) && (INPUTB != 0)

5.14 Comparison 87

Parameters

AND_In16u_←	Define for Kernel name
Out16u_K	
3	Number of ports
Port	Define for name of first input image (unsigned 16bit)
AND_KN_INA	
Port	Define for name of second input image (unsigned 16bit)
AND_KN_INB	
Port	Define for name of comparison result of the two images (unsigned 16bit).
AND_KN_OUT	

```
5.14.2.15 KERNEL_INFO apu_lower ( "apu_lower ", 3, __port(_index(0), __identifier("LOWER_KN_IN_0"), _attributes(ACF_ATTR_VEC_IN), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), _e0_size(1, 1), _ek_size(1, 1)) , _port(_index(1), _identifier("LOWER_KN_IN_1"), _attributes(ACF_ATTR_VEC_IN), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), _e0_size(1, 1), _ek_size(1, 1)) , _port(_index(2), _identifier("LOWER_KN_OUT_0"), _attributes(ACF_ATTR_VEC_OUT), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), _e0_size(1, 1), _ek_size(1, 1)) )
```

Lower kernel metadata. Compares pixelwise two unsigned 8bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA < INPUTB

Parameters

LOWER_KN	Define for Kernel name
3	Number of ports
Port LOWER_←	Define for name of first input image (unsigned 8bit)
KN_INA	
Port LOWER_←	Define for name of second input image (unsigned 8bit)
KN_INB	
Port LOWER_←	Define for name of comparison result of the two images (unsigned 8bit).
KN_OUT	

```
5.14.2.16 KERNEL_INFO apu_lower_in16s ( " apu_lower_in16s " , 3 , __port(__index(0), __identifier("LOWER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Lower kernel metadata. Compares pixelwise two signed 16bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA < INPUTB

LOWER_In16s↔	Define for Kernel name
_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (signed 16bit)
KN_INA	
Port LOWER_←	Define for name of second input image (signed 16bit)
KN_INB	
Port LOWER_←	Define for name of comparison result of the two images (unsigned 8bit).
KN_OUT	

```
5.14.2.17 KERNEL_INFO apu_lower_in32s ( "apu_lower_in32s ", 3, __port(_index(0), __identifier("LOWER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Lower kernel metadata. Compares pixelwise two signed 32bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA < INPUTB

5.14 Comparison 89

Parameters

LOWER_In32s⇔	Define for Kernel name
_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (signed 32bit)
KN_INA	
Port LOWER_←	Define for name of second input image (signed 32bit)
KN_INB	
Port LOWER_←	Define for name of comparison result of the two images. (unsigned 8bit)
KN_OUT	

```
5.14.2.18 KERNEL_INFO apu_lower_in64s ( "apu_lower_in64s " , 5 , __port(__index(0), __identifier("LOWER_KN_IN_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("LOWER_KN_IN_0_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("LOWER_KN_IN_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("LOWER_KN_IN_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(4), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Lower kernel metadata. Compares pixelwise the values of two signed 64bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA <= INPUTB

Parameters

LOWER_In64s⇔	Define for Kernel name
_KN	
3	Number of ports
Port LOWER_←	Define for name of signed 32bit high word of first signed 64bit input image
KN_INA_HIGH	
Port LOWER_←	Define for name of unsigned 32bit low word of first signed 64bit input image
KN_INA_LOW	
Port LOWER_←	Define for name of signed 32bit high word of second signed 64bit input image
KN_INB_HIGH	
Port LOWER_←	Define for name of unsigned 32bit low word of second signed 64bit input image
KN_INB_LOW	
Port LOWER_←	Define for name of unsigned 8bit comparison result of the two images.
KN_OUT	

```
5.14.2.19 KERNEL_INFO apu_lower_in64u ( "apu_lower_in64u ", 5, __port(__index(0), __identifier("LOWER_KN_IN_0_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_0_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("LOWER_KN_IN_1_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("LOWER_KN_IN_1_LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
```

Lower kernel metadata. Compares pixelwise the values of two unsigned 64bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA \leq = INPUTB

Parameters

Generated on Fri Dec 14 2018 18:20:12 for ACF/APEX Kernel SDK by Doxygen

LOWER_←	Define for Kernel name
In64u_KN	
3	Number of ports
Port LOWER_←	Define for name of unsigned 32bit high word of first unsigned signed 64bit input image
KN_INA_HIGH	
Port LOWER_←	Define for name of unsigned 32bit low word of first unsigned signed 64bit input image
KN_INA_LOW	
Port LOWER_←	Define for name of unsigned 32bit high word of second unsigned 64bit input image
KN_INB_HIGH	
Port LOWER_←	Define for name of unsigned 32bit low word of second unsigned 64bit input image
KN_INB_LOW	
Port LOWER_←	Define for name of unsigned 8bit comparison result of the two images.
KN_OUT	

```
5.14.2.20 KERNEL_INFO apu_lower_scalar ( "apu_lower_scalar ", 3, __port(_index(0), __identifier("LOWER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("LOWER_KN_Scalar"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Lower kernel metadata. Compares pixelwise an unsigned 8bit image with a scalar value. Outputs unsigned 8bit comparison result. Is true if INPUTA < INPUTB

Parameters

LOWER_SCA←	Define for Kernel name
LAR_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (unsigned 8bit)
KN_INA	
Port LOWER_←	Define for name of scalar value(unsigned 8bit)
KN_SCALAR	
Port LOWER_←	Define for name of comparison result of the two images (unsigned 8bit).
KN_OUT	

```
5.14.2.21 KERNEL_INFO apu_lowerEqual_in32s ( " apu_lowerEqual_in32s " , 3 , __port(__index(0), __identifier("LOWER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Lower or equal kernel metadata. Compares pixelwise the values of two signed 32bit images. Outputs unsigned 8bit comparison result. Is true if INPUTA <= INPUTB

LOWER_EQU⇔	Define for Kernel name
AL_In32s_KN	
3	Number of ports
Port LOWER_←	Define for name of first input image (signed 32bit)
KN_INA	

5.14 Comparison 91

Port LOWER_←	Define for name of second input image (signed 32bit)
KN_INB	
Port LOWER_←	Define for name of comparison result of the two images (unsigned 8bit).
KN_OUT	

```
5.14.2.22 KERNEL_INFO apu_mask8b ( "apu_mask8b ", 3, __port(_index(0), __identifier("MASK_IN_IMG"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MASK_IN_MASK"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("MASK_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1))))
```

MASK operator kernel metadata. Returns the values of the pixels where mask is not zero. Outputs unsigned 8bit mask result. Is IN_IMG if (MASK != 0) otherwise 0

Parameters

MASK_K	Define for Kernel name
3	Number of ports
Port MASK_K↔	Define for name of first input image (unsigned 16bit)
N_IN_IMG	
Port MASK_K↔	Define for name of second input image (unsigned 16bit)
N_IN_MASK	
Port MASK_K↔	Define for name of comparison result of the two images (unsigned 16bit).
N_OUT	

5.14.2.23 void lower (vbool * dst, vec08u * srcImage0, vec08u * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "<" operation => bool.

Comparison btw two 8bit matrices: dst[i] = (srcImage0[i] < srcImage1[i])

Parameters

dst	- [Output] boolean destination block pointer
srcImage0	- [Input] unsigned 8bit source block pointer of img 0
srcImage1	- [Input] unsigned 8bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.24 void lower_in16s (vbool * dst, vec16s * srcImage0, vec16s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 16bit "<" operation => bool.

Comparison btw two 16bit matrices: dst[i] = (srcImage0[i] < srcImage1[i])

	,
dst	- [Output] boolean destination block pointer
srcImage0	- [Input] signed 16bit source block pointer of img 0
srcImage1	- [Input] signed 16bit source block pointer of ima 1

bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.25 void lower_in32s (vbool * dst, vec32s * srcImage0, vec32s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation => bool.

Comparison btw two 32bit matrices: dst[i] = (srcImage0[i] < srcImage1[i])

Parameters

dst	- [Output] boolean destination block pointer
srcImage0	- [Input] signed 32bit source block pointer of img 0
srcImage1	- [Input] signed 32bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.26 void lower_in64s (vbool * dst, vec32s * srcImage0_high, vec32u * srcImage0_low, vec32s * srcImage1_high, vec32u * srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit "<" operation => bool.

Comparison btw two signed 64bit matrices: dst[i] = (srcImage0[i] < srcImage1[i])

Parameters

dst	- [Output] Boolean Destination block pointer
srcImage0_high	- [Input] signed 32bit high word source block pointer of img 0
srcImage0_low	- [Input] unsigned 32bit low word source block pointer of img 0
srcImage1_high	- [Input] signed 32bit high word source block pointer of img 1
srcImage1_low	- [Input] unsigned 32bit low word source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.27 void lower_in64u (vbool * dst, vec32u * srclmage0_high, vec32u * srclmage0_low, vec32u * srclmage1_high, vec32u * srclmage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 64bit "<" operation => bool.

Comparison btw two unsigned 64bit matrices: dst[i] = (srcImage0[i] < srcImage1[i])

dst	- [Output] Boolean destination block pointer
srcImage0_high	- [Input] unsigned 32bit high word of source block pointer of img 0
srcImage0_low	- [Input] unsigned 32bit low word of source block pointer of img 0

5.14 Comparison 93

srcImage1_high	- [Input] unsigned 32bit high word of source block pointer of img 1
srcImage1_low	- [Input] unsigned 32bit low word of source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.28 void lower_scalar (vec08u * dst, vec08u * srcImage, unsigned char scalar, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "<" operation => bool.

Comparison of an 8bit image with a scalar value: dst[i] = (srcImage0[i] < scalar)

Parameters

dst	- [Output] boolean destination block pointer
srcImage	- [Input] unsigned 8bit source block pointer of img 0
scalar	- [Input] unsigned 8bit source block pointer of img 1
bw	- [Input] Block width
bh	- [Input] Block height
inStrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.29 void lowerEqual_in32s (vbool * dst, vec32s * srcImage0, vec32s * srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<=" operation => bool.

Comparison btw two 32bit numbers: dst[i] = (srcImage0[i] <= srcImage1[i])

Parameters

dst	- [Output] boolean destination block pointer
srcImage0	- [Input] signed 32bit source block pointer of first image
srcImage1	- [Input] signed 32bit source block pointer of second image
bw	- [Input] Block width
bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.14.2.30 void mask_kn (vec08u * srclmage, vec08u * srclmage, vec08u * srcMask, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit mask operation => unsigned 8bit.

Elementwise MASK and image : dst[i] = (srcImage[i] if (srcMask[i]!= 0) otherwise 0

dst	- [Output] unsigned 8bit destination block pointer
srcImage	- [Input] unsigned 8bit source block pointer of image
srcMask	- [Input] unsigned 8bit source block pointer of mask
bw	- [Input] Block width

bh	- [Input] Block height
instrideW	- [Input] Source block width (in elements not bytes) including padding
outStrideW	- [Input] Destination block width (in elements not bytes) including padding

5.15 Binary descriptor matching

5.15.1 Detailed Description

Binary descriptor matching.

Collaboration diagram for Binary descriptor matching:



Functions

- KERNEL_INFO apu_match_descriptors (" apu_match_descriptors ", 5, __port(__index(0), __identifier("I↔ NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_↔ size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), ↔ __identifier("INPUT_CONFIG"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("OUTPUT_↔ 0"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(512, 1)), __port(__index(4), __identifier("OUTPUT_1"), __attributes(ACF_ATT↔ R_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_↔ size(512, 1)))
- void Match (const vec08u *apcDescriptors0, unsigned int aDescriptor0Count, const vec08u *apc
 Descriptors1, unsigned int aDescriptor1Count, int16s *apMatches0, int16s *apMatches1, int08u aThreshold, int08u aRangeCheck)

Matches binary descriptors.

5.15.2 Function Documentation

```
5.15.2.1 KERNEL_INFO apu_match_descriptors ( "apu_match_descriptors ", 5, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INPUT_CONFIG"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(512, 1)), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(512, 1)))
```

Descriptor matching kernel metadata.

MATCH DES⇔	Define for Kernel name
WIATOTI_DEO←	Define for Nemer name
CRIPTORS KN	
OHII TOHO_KIV	

5	Number of ports
Port MATCH_←	Define for name of first input descriptors array (unsigned 8bit)
DESCR_KN_IN0	
Port MATCH_←	Define for name of second input descriptors array (unsigned 8bit)
DESCR_KN_IN1	
Port	Define for name of configuration data: number of descriptors in IN0 (16-bit) number of de-
MATCH_DES⇔	scriptors in IN1 (16-bit) matching threshold (max Hamming distance) (8-bit) range check (min
CR_KN_CFG	Hamming distance between the closest and the second closest descriptors found) (8-bit)
Port	Define for name of first elements of match pairs array (unsigned 16bit)
<i>MATCH_DES</i> ↔	
CR_KN_OUT0	
Port	Define for name of second elements of match pairs array (unsigned 16bit)
<i>MATCH_DES</i> ⇔	
CR_KN_OUT1	

5.15.2.2 void Match (const vec08u * apcDescriptors0, unsigned int aDescriptor0Count, const vec08u * apcDescriptors1, unsigned int aDescriptor1Count, int16s * apMatches0, int16s * apMatches1, int08u aThreshold, int08u aRangeCheck

Matches binary descriptors.

Matches binary descriptors from group A to binary descriptors from group B. Matches with hamming distance greater than provided threshold are rejected. Ambiguous matches (i.e. the next best match's hamming distance is lesser than the provided check range) are rejected.

distance(desc0, desc1) = popcount(desc0 xor desc1)

After a descriptor desc0 (from group A) has been matched with descriptor desc1 (group B), both desc0 and desc1 are removed from further processing.

dist(desc0, desc1) has to be <= threshold, otherwise the descriptors won't be matched.

If dist(desc0, desc1A) is the smallest distance for desc0 and dist(desc0, desc1B) is the second smallest distance for desc0, the difference of these distances has to be > range check, otherwise desc0 won't be matched to anything

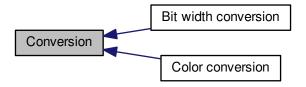
apcDescriptors0	- [Input] First input descriptor array
aDescriptor0←	- [Input] First input descriptor array size
Count	
apcDescriptors1	- [Input] Second input descriptor array
aDescriptor1←	- [Input] Second input descriptor array size
Count	
apMatches0	- [Output] First elements of match pairs
apMatches1	- [Output] Second elements of match pairs
aThreshold	- [Input] Matching threshold (max Hamming distance)
aRangeCheck	- [Input] Range check (min Hamming distance between the closest and the second closest
	descriptors found)

5.16 Conversion 97

5.16 Conversion

5.16.1 Detailed Description

Collaboration diagram for Conversion:



Modules

- Bit width conversion

 Bit width conversion.
- Color conversion

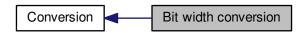
Color conversion.

5.17 Bit width conversion

5.17.1 Detailed Description

Bit width conversion.

image 16low_to_8 implementation for APEXCollaboration diagram for Bit width conversion:



Functions

- KERNEL_INFO apu_16low_to_8 (" apu_16low_to_8 ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("APU_16LOWTO8_OUT"), __attributes(ACF_ATTR_VEC_OUT),
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void f16low_to_8 (vec08u *dst, vec16u *src, int bw, int bh)
 Extracts the lower bytes.

5.17.2 Function Documentation

```
5.17.2.1 KERNEL_INFO apu_16low_to_8 ( " apu_16low_to_8 " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("APU_16LOWTO8_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Low part of 16-bit image extraction kernel metadata.

Parameters

_16_LOW_TO↔	Define for Kernel name
_8_KN	
2	Number of ports
Port_16_LOW←	Define for name of input image (unsigned 16bit)
_TO_8_KN_IN	
Port	Define for name of output image (unsigned 8bit)
_16_LOW_TO⇔	
_8_KN_OUT	

5.17.2.2 void f16low_to_8 (vec08u * dst, vec16u * src, int bw, int bh)

Extracts the lower bytes.

Extracts lower parts of the 16-bit image pixels into 8-bit image.

5.17 Bit width conversion 99

dst	- [Output] Pointer to the destination buffer
src	- [Input] Pointer to the source buffer
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block

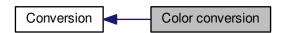
5.18 Color conversion

5.18.1 Detailed Description

Color conversion.

RGB to HSV transformation implementation for APEX.

RGB to grayscale transformation implementation for APEX.Collaboration diagram for Color conversion:



Functions

- KERNEL_INFO apu_rgb_to_grayscale (" apu_rgb_to_grayscale ", 2, __port(__index(0), __identifier("INP ← UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void rgb_to_grayscale (vec08u *apDest, const vec08u *apcSrc, int aBlockWidth, int aBlockHeight, int a
 —
 OutputSpan, int aInputSpan)

Transforms RGB to grayscale.

- KERNEL_INFO apu_rgb_to_hsv_sat ("apu_rgb_to_hsv_sat ", 2, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek← __size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_← dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_rgb_to_hsv_hue_sat (" apu_rgb_to_hsv_hue_sat ", 3, __port(_index(0), __
 identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(3, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), __attributes(ACF_ATT
 R_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
 __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_rgb_to_hsv_hue_sat_grey (" apu_rgb_to_hsv_hue_sat_grey ", 4, __port(_index(0), _
 __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(3, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_\leftarrow VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(\leftarrow index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("OUT_GREY"), __\leftarrow attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __\leftarrow ek_size(1, 1)))
- KERNEL_INFO apu_rgb_to_hsv_svr (" apu_rgb_to_hsv_svr ", 4, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek__size(1, 1)), __port(__index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT_VAL"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OUT_RED"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Transforms RGB to HSV => S.

5.18 Color conversion 101

void rgb_to_hsv_hue_sat (vec16u *apHue, vec08u *apSat, const vec08u *apcSrc, int aBlockWidth, int a
 —
 BlockHeight, int aOutputSpan, int aInputSpan)

```
Transforms RGB to HSV => (Hue,Sat)
```

• void rgb_to_hsv_hue_sat_grey (vec16u *apHue, vec08u *apSat, vec08u *grey, const vec08u *apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int aInputSpan)

```
Transforms RGB to HSV, Grey => (Hue, Sat, Grey)
```

void rgb_to_hsv_svr (vec08u *apSat, vec08u *apVal, vec08u *apRed, const vec08u *apcSrc, int aBlock
Width, int aBlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to HSV => (S, V, Red)

5.18.2 Function Documentation

```
5.18.2.1 KERNEL_INFO apu_rgb_to_grayscale ( "apu_rgb_to_grayscale ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

RGB to grayscale conversion kernel metadata.

Parameters

RGB_TO_GR↔	Define for Kernel name
AY_KN_IN	
2	Number of ports
Port RGB_TO↔	Define for name of input RGB image (unsigned 3x8bit)
_GRAY_KN_IN	
Port	Define for name of output grayscale image (unsigned 8bit)
RGB_TO_GR↔	
AY_KN_OUT	

```
5.18.2.2 KERNEL_INFO apu_rgb_to_hsv_hue_sat ( "apu_rgb_to_hsv_hue_sat ", 3, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

RGB to Hue and Saturation (of HSV) conversion kernel metadata.

RGB_HSV_KN	Define for Kernel name
3	Number of ports
Port RGB_HS↔	Define for name of input RGB image (unsigned 3x8bit)
V_KN_IN	
Port RGB_HS↔	Define for name of output SATURATION image (unsigned 8bit)
V_KN_SAT	
Port RGB_HS↔	Define for name of output HUE image (unsigned 16bit)
V_KN_HUE	

5.18.2.3 KERNEL_INFO apu_rgb_to_hsv_hue_sat_grey ("apu_rgb_to_hsv_hue_sat_grey ", 4, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), _attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OUT_GREY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

RGB to Hue, Saturation (of HSV) and to Grey conversion kernel metadata.

5.18 Color conversion 103

Parameters

RGB_HSV_H↔	Define for Kernel name
S_GREY_KN	
4	Number of ports
Port RGB_HS↔	Define for name of input RGB image (unsigned 3x8bit)
V_KN_IN	
Port RGB_HS⇔	Define for name of output SATURATION image (unsigned 8bit)
V_KN_SAT	
Port RGB_HS⇔	Define for name of output HUE image (unsigned 16bit)
V_KN_HUE	
Port RGB_HS⇔	Define for name of output Grey image (unsigned 8bit)
V_KN_GREY	

```
5.18.2.4 KERNEL_INFO apu_rgb_to_hsv_sat ( "apu_rgb_to_hsv_sat ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

RGB to HSV conversion kernel metadata.

Parameters

RGB_HSV_KN	Define for Kernel name
2	Number of ports
Port RGB_HS↔	Define for name of input RGB image (unsigned 3x8bit)
V_KN_IN	
Port RGB_HS↔	Define for name of output SATURATION image (unsigned 8bit)
V_KN_SAT	

```
5.18.2.5 KERNEL_INFO apu_rgb_to_hsv_svr ( "apu_rgb_to_hsv_svr ", 4, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT_VAL"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OUT_RED"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

RGB to Saturation, Value, Red (of HSV) conversion kernel metadata.

RGB_HSV_SV⇔	Define for Kernel name
R_KN	
4	Number of ports
Port RGB_HS↔	Define for name of input RGB image (unsigned 3x8bit)
V_KN_IN	
Port RGB_HS↔	Define for name of output SATURATION image (unsigned 8bit)
V_KN_SAT	
Port RGB_HS↔	Define for name of output VALUE image (unsigned 8bit)
V_KN_VAL	
Port RGB_HS↔	Define for name of output Red channel of the input image (unsigned 8bit)
V_KN_RED	

5.18.2.6 void rgb_to_grayscale (vec08u * apDest, const vec08u * apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to grayscale.

Transforms RGB images to grayscale images.

Parameters

apDest	- [Output] Pointer to the destination buffer
apcSrc	- [Input] Pointer to the source buffer
aBlockWidth	- [Input] Width of one data block
aBlockHeight	- [Input] Height of one data block
aOutputSpan	- [Input] Span of the destination data
alnputSpan	- [Input] Span of the source data

5.18.2.7 void rgb_to_hsv_hue_sat (vec16u * apHue, vec08u * apSat, const vec08u * apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to HSV => (Hue,Sat)

Transforms RGB images to HSV images. Returns the saturation(=(max(R,G,B)-min(R,G,B))/max(R,G,B)), the Hue + 90 deg(such that red lies at 90 degrees)

Parameters

apHue	- [Output] Pointer to the returned hue buffer (values are shifted by 90 degrees)
apSat	- [Output] Pointer to the returned saturation buffer
apcSrc	- [Input] Pointer to the source buffer (in rgb format, where rgb pixels come one after each
	other)
aBlockWidth	- [Input] Width of one data block
aBlockHeight	- [Input] Height of one data block
aOutputSpan	- [Input] Span of the destination data
alnputSpan	- [Input] Span of the source data

5.18.2.8 void rgb_to_hsv_hue_sat_grey (vec16u * apHue, vec08u * apSat, vec08u * grey, const vec08u * apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int alnputSpan)

Transforms RGB to HSV,Grey => (Hue,Sat, Grey)

Transforms RGB images to an HSV and a grey image. Returns the saturation (= (max(R,G,B)-min(R,G,B))/max(R,G,B)), the Hue + 90 deg(such that red lies at 90 degrees)

apHue	- [Output] Pointer to the returned hue buffer (values are shifted by 90 degrees)
apSat	- [Output] Pointer to the returned saturation buffer
apGrey	- [Output] Pointer to the returned grey buffer
apcSrc	- [Input] Pointer to the source buffer (in rgb format, where rgb pixels come one after each
	other)
aBlockWidth	- [Input] Width of one data block
aBlockHeight	- [Input] Height of one data block
aOutputSpan	- [Input] Span of the destination data
alnputSpan	- [Input] Span of the source data

5.18 Color conversion 105

5.18.2.9 void rgb_to_hsv_sat (vec08u * apSat, const vec08u * apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to HSV => S.

Transforms RGB images to HSV images. Returns the saturation (=(max(R,G,B)-min(R,G,B))/max(R,G,B))

Parameters 2 4 1

apSat	- [Output] Pointer to the returned saturation buffer
apcSrc	- [Input] Pointer to the source buffer (in rgb format, where rgb pixels come one after each
	other)
aBlockWidth	- [Input] Width of one data block
aBlockHeight	- [Input] Height of one data block
aOutputSpan	- [Input] Span of the destination data
alnputSpan	- [Input] Span of the source data

5.18.2.10 void rgb_to_hsv_svr (vec08u * apSat, vec08u * apVal, vec08u * apRed, const vec08u * apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int alnputSpan)

Transforms RGB to HSV = > (S,V,Red)

Transforms RGB images to HSV images. Returns the saturation (=(max(R,G,B)-min(R,G,B))/max(R,G,B)), the Value(==max(R,G,B)) and the Value(=max(R,G,B)) and the Value(=max(R,G,B)) and Value

apSat	- [Output] Pointer to the returned saturation buffer
apVal	- [Output] Pointer to the returned value buffer
apRed	- [Output] Pointer to the returned red image channel buffer
apcSrc	- [Input] Pointer to the source buffer (in rgb format, where rgb pixels come one after each
	other)
aBlockWidth	- [Input] Width of one data block
aBlockHeight	- [Input] Height of one data block
aOutputSpan	- [Input] Span of the destination data
alnputSpan	- [Input] Span of the source data

5.19 Display

5.19.1 Detailed Description

Image marking.

Collaboration diagram for Display:



Modules

· Marking on images

Image marking.

Functions

5.19.2 Function Documentation

```
5.19.2.1 KERNEL_INFO apu_mark_color_channel ( " apu_mark_color_channel " , 4 , __port(_index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_MARKER"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("INPUT_CHANNEL_INDEX"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)) )
```

Color channel marking kernel metadata.

MARK_COL_←	Define for Kernel name
KN_IN	
4	Number of ports

5.19 Display 107

Port MARK_C↔	Define for name of input image (unsigned 8bit)
OL_KN_IN	
Port	Define for name of input marker image (unsigned 8bit)
MARK_COL_←	
KN_MARKER	
Port	Define for name of color channel index (static unsigned 8bit)
MARK_COL_←	
KN_CHN_IDX	
Port MARK_C↔	Define for name of output image (unsigned 8bit)
OL_KN_OUT	

5.20 Marking on images

5.20.1 Detailed Description

Image marking.

Collaboration diagram for Marking on images:



Functions

- KERNEL_INFO apu_mark (" apu_mark ", 3, __port(_index(0), __identifier("INPUT_IMAGE"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), _
 _ek_size(1, 1)), __port(_index(1), __identifier("INPUT_MARKER"), __attributes(ACF_ATTR_VEC_IN),
 _spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2),
 _identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void mark (vec08u *dst, vec08u *srcImage, vec08u *srcMarker, int bw, int bh, int sstride, int mstride)
 Marks the image.
- void mark_color_channel (vec08u *dst, vec08u *srcImage, vec08u *srcMarker, int bw, int bh, int08u channel, int inStride, int inMarkerStride, int outStride)

Marks a color channel of the image.

5.20.2 Function Documentation

```
5.20.2.1 KERNEL_INFO apu_mark ( " apu_mark " , 3 , __port(_index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_MARKER"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image marking kernel metadata.

MARK_KN	Define for Kernel name
3	Number of ports
Port	Define for name of input image (unsigned 8bit)
MARK_KN_IN	
Port MARK_K↔	Define for name of input marker image (unsigned 8bit)
N_MARKER	
Port MARK_K↔	Define for name of output image (unsigned 8bit)
N_OUT	

5.20.2.2 void mark (vec08u * dst, vec08u * srcImage, vec08u * srcMarker, int bw, int bh, int sstride, int mstride)

Marks the image.

Marks the image. Output pixels are copied from the source image in positions where the marker pixels are zero and from the marker image otherwise.

Parameters

dst	- [Output] Pointer to the destination buffer
srcImage	- [Input] Pointer to the source buffer
srcMarker	- [Input] Pointer to the marker buffer
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block
sstride	- [Input] Input stride
mstride	- [Input] Marker stride

5.20.2.3 void mark_color_channel (vec08u * dst, vec08u * srclmage, vec08u * srclmarker, int bw, int bh, int08u channel, int inStride, int inMarkerStride, int outStride)

Marks a color channel of the image.

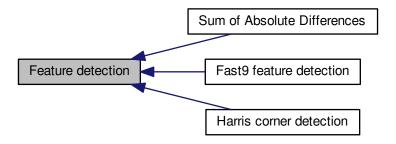
Marks the image. Output pixels are copied from the source image in positions where the marker pixels are zero and from the marker image otherwise.

dst	- [Output] Pointer to the destination buffer
srcImage	- [Input] Pointer to the source buffer
srcMarker	- [Input] Pointer to the marker buffer
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block
channel	- [Input] Index of the color channel to mark (0, 1 or 2)
inStride	- [Input] Input stride
inMarkerStride	- [Input] Marker stride
outStride	- [Input] Output stride

5.21 Feature detection

5.21.1 Detailed Description

Collaboration diagram for Feature detection:



Modules

· Fast9 feature detection

FAST9 feature point detection.

· Harris corner detection

Harris feature point detection.

• Sum of Absolute Differences

SAD (Sum of Absolute Differences \sim = image similarity)

5.22 Fast9 feature detection 111

5.22 Fast9 feature detection

5.22.1 Detailed Description

FAST9 feature point detection.

Collaboration diagram for Fast9 feature detection:



Functions

- KERNEL_INFO apu_fast9 (" apu_fast9 ", 3, __port(__index(0), __identifier("IN_Img"), __attributes(ACF_ ← ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), _ ← __port(__index(1), __identifier("OUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("IN_Thr"), __ ← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(1, 1)))
- void apu_fast9_unsuppressed_score (const vec08u *apcSrc, vec08u *apDst, int aSourceStride, int a
 — DestinationStride, int aBlockWidth, int aBlockHeight, int08u aThreshold)

FAST9 corner detection.

5.22.2 Function Documentation

```
5.22.2.1 KERNEL_INFO apu_fast9 ( "apu_fast9 ", 3 , __port(__index(0), __identifier("IN_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("IN_Thr"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

FAST9 feature point detection kernel metadata.

FAST9_KN	Define for Kernel name
3	Number of ports
Port	Define for name of input image (unsigned 8bit)
FAST9_KN_IN0	
Port 'FAST9_K↔	Define for name of output image (unsigned 8bit)
N_OUT	
Port	Define for name of threshold used for classifying ring pixels (unsigned 8bit)
FAST9_KN_IN1	(brighter/darker/similar)

5.22.2.2 void apu_fast9_unsuppressed_score (const vec08u * apcSrc, vec08u * apDst, int aSourceStride, int aDestinationStride, int aBlockWidth, int aBlockHeight, int08u aThreshold)

FAST9 corner detection.

Finds the corners in the input data using the FAST9 algorithm. Outputs corner scores or 0 if not a corner.

For each input pixel a 16-pixel circle centered at the processed pixel is considered. The circle pixels are classified as darker, brighter or similar to the central pixel depending on the provided threshold. The central pixel is considered as a corner if and only if there is a contiguous segment of 9 pixels which are all classified as brighter or darker in the circle.

See http://www.edwardrosten.com/work/fast.html

apcSrc	- [Input] pointer to the source buffer
apcDst	- [Output] pointer to the destination buffer
aSourceStride	- [Input] line stride of the source data
aDestination←	- [Input] line stride of the destination data
Stride	
aBlockWidth	- [Input] width of one data block
aBlockHeight	- [Input] height of one data block
aThreshold	- [Input] threshold used for pixel classification

5.23 Harris corner detection 113

5.23 Harris corner detection

5.23.1 Detailed Description

Harris feature point detection.

Collaboration diagram for Harris corner detection:



Functions

- KERNEL_INFO apu_harris (" apu_harris ", 7, __port(__index(0), __identifier("INPUT_GX"), __attributes(A \leftarrow CF_ATTR_VEC_IN), __spatial_dep((4 >> 1),(4 >> 1),(4 >> 1),(4 >> 1)), __e0_data_type(d16s), __ \leftarrow e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_GY"), __attributes(ACF_ATTR_V \leftarrow EC_IN), __spatial_dep((4 >> 1),(4 >> 1),(4 >> 1),(4 >> 1)), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INPUT_K_RBS_WINDOW"), __attributes(ACF_ATTR_S \leftarrow CL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("OUTPUT_RESPONSE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_ \leftarrow dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("O \leftarrow UTPUT_X2TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2 *(4 >> 1), 16+2 *(4 >> 1))), __port(_ \leftarrow _index(5), __identifier("OUTPUT_Y2TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIX \leftarrow ED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2 *(4 >> 1), 16+2 *(4 >> 1), 16+2 *(4 >> 1))), __port(__ \leftarrow OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0, 0, __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2 *(4 >> 1), 16+2 *(4 >> 1))), __port(__ \leftarrow OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0, 0, __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2 *(4 >> 1), 16+2 *(4 >> 1))), __ek_size(16+2 *(4 >> 1))))
- void apuHarrisResponse (vec16u *apResponse, vec16s *apGradX, vec16s *apGradY, vec16s *apXSqrTmp, vec16s *apXYTemp, int aBlockWidth, int aBlockHeight, int aStride, int k, int response BitShift, int aWindowSize, int thresh, bool isFirstSlice)

Harris corner detection.

- KERNEL_INFO apu_harris (" apu_harris ", 5, __port(_index(0), __identifier("INPUT_GX"), __attributes(A \(\capstaller \) CF_ATTR_VEC_IN), __spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_GY"), __attributes(ACF_ATTR_VEC_IN), \(\capstaller \) _spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INPUT_K_RBS_WINDOW"), __attributes(ACF_ATTR_SCL_IN_STATIC_\(\capstaller \) FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("OUTPUT_RESPONSE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("OUTPUT_\(\capstaller \) TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE), __\(\capstaller \) e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- void apuHarrisResponse (vec16u *apResponse, vec16s *apGradX, vec16s *apGradY, vec16s *apTemp, int aBlockWidth, int aBlockHeight, int aStride, int k, int responseBitShift, int aWindowSize, int thresh, bool is← FirstSlice)

Harris corner detection.

5.23.2 Function Documentation

5.23.2.1 KERNEL_INFO apu_harris ("apu_harris ", 5 , __port(__index(0), __identifier("INPUT_GX"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_GY"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("INPUT_K_RBS_WINDOW"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("OUTPUT_RESPONSE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(4), __identifier("OUTPUT_TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, HARRIS_HALF_WINDOW_SIZE, __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))

Harris corner detection kernel metadata.

Parameters

HARRIS_KN	Define for Kernel name
4	Number of ports
Port HARRIS_←	Define for name of gradient X component input image (signed 16bit)
KN_IN_GX	
Port HARRIS_←	Define for name of gradient Y component input image (signed 16bit)
KN_IN_GY	
Port	Define for name of detector sensitivity, response bit shift and window size (unsigned 16bit)
HARRIS_KN_I⊷	
N_K_RBS_WIN	
Port HARRIS_←	Define for name of output image (unsigned 16bit)
KN_OUT_RESP	
Port HARRIS_←	Define for name of temporary buffer (signed 16bit)
KN_OUT_TEM⇔	
P_BUF	

```
5.23.2.2 KERNEL_INFO apu_harris ( "apu_harris ", 7, __port(_index(0), __identifier("INPUT_GX"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep((4 >> 1),(4 >> 1),(4 >> 1),(4 >> 1)), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_GY"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep((4 >> 1),(4 >> 1),(4 >> 1)), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("INPUT_K_RBS_WINDOW"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)) , __port(_index(3), __identifier("OUTPUT_RESPONSE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(4), __identifier("OUTPUT_X2TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2*(4 >> 1), 16+2*(4 >> 1))) , __port(_index(5), __identifier("OUTPUT_X2TEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2*(4 >> 1), 16+2*(4 >> 1))) , __port(_index(6), __identifier("OUTPUT_XYTEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2*(4 >> 1), 16+2*(4 >> 1))) , __port(_index(6), __identifier("OUTPUT_XYTEMP_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(16+2*(4 >> 1), 16+2*(4 >> 1))) )
```

Harris corner detection kernel metadata.

HARRIS_KN	Define for Kernel name

7	Number of ports
Port HARRIS_←	Define for name of gradient X component input image (signed 16bit)
KN_IN_GX	
Port HARRIS_←	Define for name of gradient Y component input image (signed 16bit)
KN_IN_GY	
Port	Define for name of detector sensitivity, response bit shift and window size (unsigned 16bit)
HARRIS_KN_I⊷	
N_K_RBS_WIN	
Port HARRIS_←	Define for name of output image (unsigned 16bit)
KN_OUT_RESP	
Port HARRIS_←	Define for name of temporary buffer holding gx*gx values (signed 16bit)
KN_OUT_X2T↔	
EMP_BUF	
Port HARRIS_←	Define for name of temporary buffer holding gy∗gy values (signed 16bit) [∧] M
KN_OUT_Y2T↔	
EMP_BUF	
Port HARRIS_←	Define for name of temporary buffer holding gx∗gy values (signed 16bit) [∧] M
KN_OUT_XYT↔	
EMP_BUF	

5.23.2.3 void apuHarrisResponse (vec16u * apResponse, vec16s * apGradX, vec16s * apGradY, vec16s * apTemp, int aBlockWidth, int aBlockHeight, int aStride, int k, int responseBitShift, int aWindowSize, int thresh, bool isFirstSlice)

Harris corner detection.

Finds the corners in the input data using the Harris algorithm. Outputs a Harris response value for each pixel.

out[i] = $(\det(A[i]) - k * trace(A[i])^2) >> responseBitShift$ where A[i] is a structure tensor for pixel i

See $\label{lem:http://docs.opencv.org/doc/tutorials/features2d/trackingmotion/harris_{} \Leftrightarrow detector/harris_detector.html$

Parameters

apResponse	- output Harris response image
	· · · · · · · · · · · · · · · · · · ·
apGradX	- image gradient x component
apGradY	- image gradient y component
aBlockWidth	- CU block width
aBlockHeight	- CU block height
aStride	- horizontal gradient image stride
k	- Harris detector sensitivity
responseBit⇔	bit shift that will be applied to the output response
Shift-	
aWindowSize	- Harris detector window size
thresh	- threshold, below which corner values are set to zero
isFirstSlice	- if we are at first slice, initialization is performed

5.23.2.4 void apuHarrisResponse (vec16u * apResponse, vec16s * apGradX, vec16s * apGradY, vec16s * apXSqrTmp, vec16s * apXYTemp, int aBlockWidth, int aBlockHeight, int aStride, int k, int responseBitShift, int aWindowSize, int thresh, bool isFirstSlice)

Harris corner detection.

Finds the corners in the input data using the Harris algorithm. Outputs a Harris response value for each pixel.

out[i] = $(\det(A[i]) - k * trace(A[i])^2) >> responseBitShift$ where A[i] is a structure tensor for pixel i

See $\label{lem:http://docs.opencv.org/doc/tutorials/features2d/trackingmotion/harris_ control contro$

apResponse	- output Harris response image
apGradX	- image gradient x component
apGradY	- image gradient y component
apXSqrTmp	- temporary buffer to store gradX^2 values on overlapping zones
apYSqrTmp	- temporary buffer to store gradY [∧] 2 values on overlapping zones
apXYTmp	- temporary buffer to store gradX*gradY values on overlapping zones
aBlockWidth	- CU block width
aBlockHeight	- CU block height
aStride	- horizontal gradient image stride
k	- Harris detector sensitivity
responseBit←	bit shift that will be applied to the output response
Shift-	
aWindowSize	- Harris detector window size
thresh	- threshold, below which corner values are set to zero
isFirstSlice	- if we are at first slice, initialization is performed

5.24 Sum of Absolute Differences

5.24.1 Detailed Description

SAD (Sum of Absolute Differences ~= image similarity)

Collaboration diagram for Sum of Absolute Differences:



Functions

- KERNEL_INFO apu_sad (" apu_sad ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_A← TTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 4)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(8, 8)), __port(__index(2), __identifier("OUTPUT_← 0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_← size(4, 1), __ek_size(1, 1)))
- void apu_sad_impl (vec08u *lpvln0, int16_t lStrideln0, int16_t lChunkWidthln0, int16_t lChunkHeight
 In0, vec08u *lpvln1, int16_t lStrideln1, int16_t lChunkWidthln1, int16_t lChunkHeightln1, vec32u *lpvOut0,
 int16 t lStrideOut0, int16 t lChunkWidthOut0, int16 t lChunkHeightOut0)

Sum of absolute differences. Store the minimum of all differences and the location of the minimum in image0.

5.24.2 Function Documentation

```
5.24.2.1 KERNEL_INFO apu_sad ( "apu_sad ", 3, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 4)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(8, 8)), __port(_index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(4, 1), __ek_size(1, 1)) )
```

SAD Sum of Absolute Differences kernel metadata.

SAD_KN	Definition for kernel name
3	Number of ports
Port	Define for name of first input image (unsigned 8bit)
SAD_KN_IN0	
Port	Define for name of second input image (unsigned 8bit)
SAD_KN_IN1	
Port	Define for name of output image containing the SAD values for each pixel (unsigned 8bit)
SAD KN OUT	

5.24.2.2 void apu_sad_impl (vec08u * *IpvIn0*, int16_t *IStrideIn0*, int16_t *IChunkWidthIn0*, int16_t *IChunkHeightIn0*, vec08u * *IpvIn1*, int16_t *IStrideIn1*, int16_t *IChunkWidthIn1*, int16_t *IChunkHeightIn1*, vec32u * *IpvOut0*, int16_t *IStrideOut0*, int16_t *IChunkWidthOut0*, int16_t *IChunkHeightOut0*)

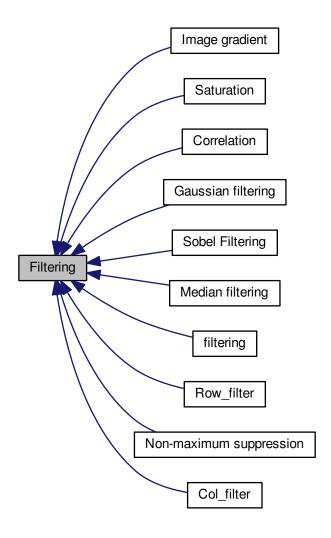
Sum of absolute differences. Store the minimum of all differences and the location of the minimum in image0.

lpvIn0	- [Input] pointer to the first source buffer
lStrideIn0	- [Input] line stride of the first source buffer
IChunkWidthIn0	- [Input] width of one data block of the first source buffer
IChunkHeightIn0	- [Input] height of one data block of the first source buffer
lpvln1	- [Input] pointer to the second source buffer
lStrideIn1	- [Input] line stride of the second source buffer
IChunkWidthIn1	- [Input] width of one data block of the second source buffer
IChunkHeightIn1	- [Input] height of one data block of the second source buffer \
lpvOut0	- [Output] pointer to the destination buffer
IStrideOut0	- [Input] line stride of the destination data
IChunkWidth⇔	- [Input] width of one data block of the destination buffer
Out0	
IChunkHeight⇔	- [Input] height of one data block of the destination buffer
Out0	

5.25 Filtering

5.25.1 Detailed Description

Collaboration diagram for Filtering:



Modules

· Col_filter

Filter an image row with a column_filter (i.e. Matrix * Col_Vector multiplication)

Correlation

Correlation, respectively convolution with a reversed filter.

filtering

General Filtering.

· Median filtering

Median filtering.

Sobel Filtering

5.25 Filtering 121

Sobel filtering.

· Gaussian filtering

Gaussian filtering.

• Image gradient

Image gradient.

• Non-maximum suppression

Non-maximum suppression.

• Row_filter

Filter an image column with a row_filter(i.e. Row_Vector * Matrix multiplication)

• Saturation

Saturation.

5.26 Col filter

5.26.1 Detailed Description

Filter an image row with a column_filter (i.e. Matrix * Col_Vector multiplication)
Collaboration diagram for Col_filter:



Functions

- KERNEL_INFO col_filter (" col_filter ", 3, __port(__index(0), __identifier("SRC"), __attributes(ACF_ATTR → __VEC_IN), __spatial_dep(COL_PADDING, COL_PADDING, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_ ← FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_COLS, 1)), __port(__index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __← e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void col_filter (vec08u *dst, int dstStride, const vec08u *src, int srcStride, int rows, int cols, const unsigned char *filter, int filterSize, int filterQ)

Apply a column filter to an image.

Variables

- const int FILTER COLS = 3
- const int COL_PADDING = FILTER_COLS >> 1

5.26.2 Function Documentation

```
5.26.2.1 KERNEL_INFO col_filter ( " col_filter " , 3 , __port(__index(0), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(COL_PADDING, COL_PADDING, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_COLS, 1)) , __port(__index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Column filter kernel metadata.

COL_FILTER_←	Define for Kernel name
KN	
3	Number of ports
Port COL_FIL↔	Define for name of input image (unsigned 8bit)
TER_KN_SRC	

5.26 Col_filter 123

Port	Define for name of filter coefficients(unsigned 8bit)
COL_FILTER_←	
KN_COEFFS	
Port COL_FIL↔	Define for name of output image (unsigned 8bit)
TER_KN_DST	

5.26.2.2 void col_filter (vec08u * dst, int dstStride, const vec08u * src, int srcStride, int rows, int cols, const unsigned char * filter, int filterSize, int filterQ)

Apply a column filter to an image.

First we decrement the number of fractional bits in anticipation of rounding the final result.

Next, we offset the source pointer so that it points to the beginning of the padded region of the source image. The offset amount is the number of padded columns (see The Column Filter).

For each pixel, we accumulate the weighted sum in a 16-bit register to avoid overflow. When casting the weighted sum to an integer we round the value. This means we right bit shift the sum by 1 less the fractional bits, add 1, then right shift one more.

Parameters

dst	Pointer to the destination image.
dstStride	Stride of the destination image.
src	Pointer to the source image. The source image is assumed to be padded according to the
	filter size. However, src points the top left corner of the <i>unpadded</i> image region.
srcStride	Stride of the padded source image.
rows	Rows of the source image, not including padding.
cols	Columns of the source image, not including padding.
filter	Pointer to column filter coefficients. The coefficients are unsigned 8-bit fixed point numbers.
filterSize	Size of the column filter. The size must be odd so that the filter is centered about the pixel.
filterQ	Number of fractional bits for the filter coefficients.

5.26.3 Variable Documentation

5.26.3.1 const int COL_PADDING = FILTER_COLS >> 1

The number of padded columns that must be added to both sides of the source image.

5.26.3.2 const int FILTER_COLS = 3

The number of columns (i.e. size) of the column filter.

5.27 Correlation

5.27.1 Detailed Description

Correlation, respectively convolution with a reversed filter.

Collaboration diagram for Correlation:



Macros

- #define inputFiltUpScale 3
- #define scharrFiltUpscale 1

Typedefs

typedef void(* corrKernelPtr)(vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Functions

- KERNEL_INFO apu_gradient_x (" apu_gradient_x ", 2, __port(__index(0), __identifier("INPUT_Img"), _
 _ attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_gradient_y (" apu_gradient_y ", 2, __port(__index(0), __identifier("INPUT_Img"), _
 _attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_scharr_x (" apu_scharr_x ", 2, __port(_index(0), __identifier("INPUT_Img"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_scharr_y (" apu_scharr_y ", 2, __port(_index(0), __identifier("INPUT_Img"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation (" apu_correlation ", 5, __port(__index(0), __identifier("INPUT_Img"), \(\to \) attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), \(\to \) spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(__index(3), __identifier("INPU\(\to \) T_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_\(\to \) data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __\(\to \) e0_size(1, 1), __ek_size(1, 1)))

5.27 Correlation 125

KERNEL_INFO apu_correlation_1x3 (" apu_correlation_1x3 ", 5, __port(__index(0), __identifier("INPU ← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V ← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__ ← index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 3)), __port(__index(3), __identifier("INPUT ← Index(3), __identifier("Index(3), __identifier("Index(3

- KERNEL_INFO apu_correlation_3x1 (" apu_correlation_3x1 ", 5, __port(__index(0), __identifier("INPU← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 0, 0), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 1)), __port(_index(3), __identifier("INPUT← __tilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data← __type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFI"), __← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_← size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_3x3 (" apu_correlation_3x3 ", 5, __port(__index(0), __identifier("INPU← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(__index(3), __identifier("INPUT← __FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data← __type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_← size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_5x5 (" apu_correlation_5x5 ", 5, __port(_index(0), __identifier("INPU \cup T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_\cup size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V \cup EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__\cup index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(5, 5)), __port(__index(3), __identifier("INPUT \cup Index(2), __itentifier("INPUT \cup Index(3), __itentifier("INPUT \cup Index(4), __itentifier("Index(4), __iten
- KERNEL_INFO apu_correlation_7x7 (" apu_correlation_7x7 ", 5, __port(__index(0), __identifier("INPU← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(7, 7)), __port(__index(3), __identifier("INPUT← __FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data← __type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_← size(1, 1), __ek_size(1, 1)))
- · void initFilters ()

Initializes the array of filter function pointers.

void performCorrelation (int16u filterFlags, vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s *filter← Coefs)

Computes time optimized the correlation with a general filter.

void correlation_filter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

void correlation__antisymXfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric X filter.

void correlation__symXfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric X filter.

void correlation__symYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric Y filter.

void correlation__antisymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric Y filter.

• void correlation_symXYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric in X and Y filter.

void correlation_symXantisymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter← Coefs)

Computes the correlation with a symmetric in X and anti-symmetric in Y filter.

void correlation_symXantisymYfilter_16s (vec16s *dst, vec16s *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s *filter← Coefs)

Computes the 16bit correlation with a symmetric in X and anti-symmetric in Y filter.

void correlation__antisymXsymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter← Coefs)

Computes the correlation with an anti-symmetric in X and symmetric in Y filter.

void correlation__antisymXsymYfilter_16s (vec16s *dst, vec16s *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter← Coefs)

Computes the 16bit correlation with an anti-symmetric in X and symmetric in Y filter.

void correlation__antisymXYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric in X and anti-symmetric in Y filter.

5.27.2 Function Documentation

```
5.27.2.1 KERNEL_INFO apu_correlation ( "apu_correlation ", 5, __port(__index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(__index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a 1D or 2D filter. Outputs correlation result (signed 16bit).

Warning

The filter width and height (max 11x11) have to be defined in the macro definitions FW and FH for the currently used kernel!

5.27 Correlation 127

Parameters

CORR_KN	Define for Kernel name
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH >> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	
Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel⊷	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.2 KERNEL_INFO apu_correlation_1x3 ( "apu_correlation_1x3 ", 5, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 3)), __port(_index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC - FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFl"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a filter of size 1x3 (convenience kernel for CORR_K). Outputs correlation result (signed 16bit).

Warning

The filter width and height have to be defined in the macro definitions FW and FH for the currently used kernel!

CORR_KN_←	Define for Kernel name
1x3_KN	
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH>> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	

Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel⊷	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.3 KERNEL_INFO apu_correlation_3x1 ( "apu_correlation_3x1 ", 5, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 1)), __port(_index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFI"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a filter of size 3x1 (convenience kernel for CORR_K). Outputs correlation result (signed 16bit).

Warning

The filter width and height have to be defined in the macro definitions FW and FH for the currently used kernel!

Parameters

CORR_KN_←	Define for Kernel name
3x1_KN	
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH >> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	
Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel⊷	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.4 KERNEL_INFO apu_correlation_3x3 ( "apu_correlation_3x3 ", 5, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(_index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC __FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFI"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a filter of size 3x3 (convenience kernel for CORR_K). Outputs correlation result (signed 16bit).

Warning

The filter width and height have to be defined in the macro definitions FW and FH for the currently used kernel!

5.27 Correlation 129

Parameters

CORR_KN_←	Define for Kernel name
3x3_KN	
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH >> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	
Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel⊷	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.5 KERNEL_INFO apu_correlation_5x5 ( "apu_correlation_5x5 ", 5, __port(__index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(5, 5)), __port(__index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a filter of size 5x5 (convenience kernel for CORR_K). Outputs correlation result (signed 16bit).

Warning

The filter width and height have to be defined in the macro definitions FW and FH for the currently used kernel!

CORR_KN_←	Define for Kernel name
5x5_KN	
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH>> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	

Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel←	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.6 KERNEL_INFO apu_correlation_7x7 ( "apu_correlation_7x7 ", 5, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(7, 7)), __port(_index(3), __identifier("INPUT_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC -FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

General correlation kernel metadata. General correlation of input unsigned 8bit image with a filter of size 7x7 (convenience kernel for CORR_K). Outputs correlation result (signed 16bit).

Warning

The filter width and height have to be defined in the macro definitions FW and FH for the currently used kernel!

Parameters

CORR_KN_←	Define for Kernel name
7x7_KN	
5	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit). The spatial dependencies of this port - HFW
Kernel_Input	and HFH - are half the filter width/height. They are defined in the header of this file as HFW
	FW>>1 and HFH FH>> 1
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	
Port CORR_←	Define for name of filter coefficients the input has to be convolved with (signed 16bit)
Kernel_Filter	
Port	Define for name of the scalar value of the normalization factor used for the filter (signed 16bit)
CORR_Kernel⊷	
_FilterScale	
Port	Define for name of the scalar value of the symmetry flag defined for this filter as it is specified
CORR_Kernel⊷	in "symmetry_flags.h" file (unsigned 16bit)
_FiltSymmFl	

```
5.27.2.7 KERNEL_INFO apu_gradient_x ( " apu_gradient_x " , 2 , __port(__index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Gradient in X direction kernel metadata. Convolution of input unsigned 8bit image with a [-1 0 1] row-filter. Outputs correlation result (signed 16bit) (i.e. the gradients in X direction of the input image)

CORR_GRAD↔	Define for Kernel name
_X_KN	

5.27 Correlation 131

2	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit)
Kernel_Input	
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	

```
5.27.2.8 KERNEL_INFO apu_gradient_y ( " apu_gradient_y " , 2 , __port(__index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Gradient in Y direction kernel metadata. Convolution of input unsigned 8bit image with a $[-1\ 0\ 1]^T$ column-filter. Outputs correlation result (signed 16bit) (i.e. the gradients in Y direction of the input image)

Parameters

CORR_GRAD↔	Define for Kernel name
_Y_KN	
2	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit)
Kernel_Input	
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	

```
5.27.2.9 KERNEL_INFO apu_scharr_x ( " apu_scharr_x " , 2 , __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Gradient in X direction kernel metadata for Scharr_X filter. Convolution of input unsigned 8bit image with a scharr_x filter. Outputs correlation result (signed 16bit) (i.e. the gradients in X direction of the input image)

Parameters

CORR_SCHA↔	Define for Kernel name
RR_X_KN	
2	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit)
Kernel_Input	
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	

```
5.27.2.10 KERNEL_INFO apu_scharr_y ( " apu_scharr_y " , 2 , __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Gradient in Y direction kernel metadata for Scharr_Y filter. Convolution of input unsigned 8bit image with a scharr_y filter. Outputs correlation result (signed 16bit) (i.e. the gradients in Y direction of the input image)

Parameters

Generated on Fri Dec 14 2018 18:20:12 for ACF/APEX Kernel SDK by Doxygen

CORR_SCHA⇔	Define for Kernel name
RR_Y_KN	
2	Number of ports
Port CORR_←	Define for name of input image (unsigned 8bit)
Kernel_Input	
Port CORR_←	Define for name of correlation result (signed 16bit)
Kernel_Output	

5.27.2.11 void correlation_antisymXfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with an anti-symmetric X filter.

Calculate the correlation with a filter, which is anti-symmetric in X direction.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.12 void correlation_antisymXsymYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with an anti-symmetric in X and symmetric in Y filter.

Calculate the correlation with a filter, which is anti-symmetric in X and symmetric in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.13 void correlation_antisymXsymYfilter_16s (vec16s * dst, vec16s * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the 16bit correlation with an anti-symmetric in X and symmetric in Y filter.

5.27 Correlation 133

Calculate the correlation of a signed 16bit image with a filter, which is anti-symmetric in X and symmetric in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.14 void correlation_antisymXYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with an anti-symmetric in X and anti-symmetric in Y filter.

Calculate the output of correlation with a filter, which is anti-symmetric in X and anti-symmetric in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.15 void correlation_antisymYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with an anti-symmetric Y filter.

Calculate the correlation with a filter, which is anti-symmetric in Y direction.

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output

xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.16 void correlation_symXantisymYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with a symmetric in X and anti-symmetric in Y filter.

Calculate the correlation with a filter, which is symmetric in X and anti-symmetric in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.17 void correlation_symXantisymYfilter_16s (vec16s * dst, vec16s * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s * filterCoefs)

Computes the 16bit correlation with a symmetric in X and anti-symmetric in Y filter.

Calculate the correlation of a signed 16bit image with a filter, which is symmetric in X and anti-symmetric in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.18 void correlation_symXfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s sstr, int16s ss

Computes the correlation with a symmetric X filter.

Calculate the correlation with a filter, which is symmetric in X direction.

5.27 Correlation 135

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.19 void correlation_symXYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s

Computes the correlation with a symmetric in X and Y filter.

Calculate the correlation with a filter, which is symmetric in X and in Y directions.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.20 void correlation_symYfilter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Computes the correlation with a symmetric Y filter.

Calculate the correlation with a filter, which is symmetric in Y direction.

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)

filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.21 void correlation_filter (vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s * filterCoefs)

Calculate the output of correlation with a general filter.

Parameters

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)
filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.27.2.22 void initFilters ()

Initializes the array of filter function pointers.

Initializes the array of filter function pointers, to correspond to the right filters. Should be called once, before filtering.

5.27.2.23 void performCorrelation (int16u filterFlags, vec16s * dst, vec08u * src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s * filterCoefs)

Computes time optimized the correlation with a general filter.

Based on the symmetry flag, this function chooses automatically the correct filter kernel and calculates the output of correlation with a general filter.

filterFlags	- [Input] flags indicating the symmetry of the filter coefficients. For possible values see the
	definitions in this file
dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height
destBw	- [Input] Block width (including stride) of the output
xSkip,ySkip	- [Input] number of pixels to be skipped in x/y direction before convolving again, i.e. if desti-
	nation has to be directly downsampled (=1 for no downsampling)

5.27 Correlation 137

filtWidth	- [Input] the x-size of the filter
filtHeight	- [Input] the y-size of the filter
filtScaling	- [Input] the factor to divide the output of the filter with
filterCoefs	- [Input] the coefficients of the filter

5.28 filtering

5.28.1 Detailed Description

General Filtering.

Collaboration diagram for filtering:



Functions

- KERNEL_INFO apu_filter_a (" apu_filter_a ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(A CF_ATTR_VEC_IN), __spatial_dep(3 >> 1, 3 >> 1, 3 >> 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_COEF"), __attributes(ACF_ATTR_SCL_IN_SCTATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(3 *3, 1)), _comport(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void apu_filter_a_impl (vec08u *src, int16_t sstr, uint8_t *coef, vec08u *dst, int16_t dstr, int16_t bw, int16_t bh)

General filtering function with a 1D/2D-filter.

5.28.2 Function Documentation

5.28.2.1 KERNEL_INFO apu_filter_a ("apu_filter_a ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR \leftarrow _VEC_IN), __spatial_dep(3 >> 1, 3 >> 1, 3 >> 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_COEF"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(3 *3, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

General filtering function with a 2D-filter

Parameters

FILTER_A_KN	Define for Kernel name
3	Number of ports
Port FILTER_←	Define for name of input image (unsigned 8bit)
A_KN_IN	
Port	Define for name of filter coefficients (totally FILTER_W * FILTER_H coefficients are allowed
FILTER_A_KN↔	right now) (unsigned 8bit)
_IN_COEF	
Port FILTER_←	Define for name of filtering result (unsigned 8bit)
A_KN_OUT	

5.28.2.2 void apu_filter_a_impl (vec08u * src, int16_t sstr, uint8_t * coef, vec08u * dst, int16_t dstr, int16_t bw, int16_t bh)

General filtering function with a 1D/2D-filter.

5.28 filtering 139 Calculate the output of convolution with a general filter.

src	- [Input] Source block pointer
sstr	- [Input] Source block stride
coef	- [Input] The filter coefficients
dst	- [Output] Destination block pointer
dstr	- [Input] Destination block stride
bw	- [Input] Block width
bh	- [Input] Block height

5.29 Median filtering 141

5.29 Median filtering

5.29.1 Detailed Description

Median filtering.

Collaboration diagram for Median filtering:



Functions

KERNEL_INFO median_3x3_8bpp (" median_3x3_8bpp ", 2, __port(__index(0), __identifier("DST"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Filtering with a Median 3x3-filter.

• void apu_flt_median_3x3 (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

Filter with a 3x3 median filter.

5.29.2 Function Documentation

5.29.2.1 void apu_fit_median_3x3 (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Filter with a 3x3 median filter.

Parameters

dst	- [Output] Destination block pointer
dstr	- [Input] Destination block stride
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height

Filtering with a Median 3x3-filter.

Filtering with a Median 3x3-filter

MEDIAN_3X3↔	Define for Kernel name
_8BPP_KN	
2	Number of ports
Port MEDIAN_←	Define for name of input image (unsigned 8bit)
3X3_8BPP_IN	
Port	Define for name of filtering result (unsigned 8bit)
MEDIANL_3↔	
X3_8BPP_OUT	

5.30 Sobel Filtering 143

5.30 Sobel Filtering

5.30.1 Detailed Description

Sobel filtering.

Collaboration diagram for Sobel Filtering:



Functions

KERNEL_INFO sobel_3x3_8bpp (" sobel_3x3_8bpp ", 2, __port(__index(0), __identifier("SRC"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

- void apu_flt_sobel_3x3_x (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)
 - Calculate first order derivative x using sobel 3x3.
- void apu_flt_sobel_3x3_y (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh) Calculate first order derivative y using sobel 3x3.
- void apu_flt_sobel_3x3 (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

5.30.2 Function Documentation

5.30.2.1 void apu_flt_sobel_3x3 (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

Parameters

dst	- [Output] Destination block pointer
dstr	- [Input] Destination block stride
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height

5.30.2.2 void apu_flt_sobel_3x3_x (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Calculate first order derivative x using sobel 3x3.

Parameters

dst	- [Output] Destination block pointer
dstr	- [Input] Destination block stride
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height

5.30.2.3 void apu_flt_sobel_3x3_y (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Calculate first order derivative y using sobel 3x3.

Parameters

dst	- [Output] Destination block pointer
dstr	- [Input] Destination block stride
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height

```
5.30.2.4 KERNEL_INFO sobel_3x3_8bpp ( " sobel_3x3_8bpp " , 2 , __port(__index(0), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

SOBEL_3X3_←	Define for Kernel name
8BPP_KN	
2	Number of ports
Port SOBEL_3⇔	Define for name of input image (unsigned 8bit)
X3_8BPP_IN	
Port SOBEL_3↔	Define for name of filtering result (unsigned 8bit)
X3_8BPP_OUT	

5.31 Gaussian filtering 145

5.31 Gaussian filtering

5.31.1 Detailed Description

Gaussian filtering.

Collaboration diagram for Gaussian filtering:



Functions

- KERNEL_INFO apu_gauss_3x3 (" apu_gauss_3x3 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void apu_gauss_3x3 (vec08u *apOut, const vec08u *apcIn, int aOutStride, int aInStride, int aTileWidth, int aTileHeight)

3x3 gaussian filter.

- KERNEL_INFO apu_gauss_5x5 (" apu_gauss_5x5 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void Gauss_5x5__filter (vec08u *dst, vec08u *src, int sstr, int bw, int bh)

5x5 gaussian filter.

5.31.2 Function Documentation

```
5.31.2.1 KERNEL_INFO apu_gauss_3x3 ( "apu_gauss_3x3 " , 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

3x3 Gaussian blur kernel metadata.

<i>GAUSS_3x3</i> _←	Define for Kernel name
KN	
2	Number of ports
Port GAUSS_←	Define for name of input image (unsigned 8bit)
3x3_KN_IN	
Port GAUSS_←	Define for name of blurred output image (unsigned 8bit)
3x3_KN_OUT	

5.31.2.2 void apu_gauss_3x3 (vec08u * apOut, const vec08u * apcIn, int aOutStride, int aInStride, int aTileWidth, int aTileHeight)

3x3 gaussian filter.

Calculate the output of 3x3 gaussian filter.

Parameters 4 8 1

apOut	- [Output] pointer to the destination buffer
apcIn	- [Input] pointer to the source buffer
aOutStride	- [Input] line stride of the destination data
alnStride	- [Input] line stride of the source data
aTileWidth	- [Input] width of one data block
aTileHeight	- [Input] height of one data block

Returns

number of corners found

```
5.31.2.3 KERNEL_INFO apu_gauss_5x5 ( " apu_gauss_5x5 " , 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

5x5 Gaussian blur kernel metadata.

Parameters

<i>GAUSS_5x5_</i> ↔	Define for Kernel name
KN	
2	Number of ports
Port GAUSS_←	Define for name of input image (unsigned 8bit)
5x5_KN_IN	
Port GAUSS_←	Define for name of blurred output image name (unsigned 8bit)
5x5_KN_OUT	

5.31.2.4 void Gauss_5x5__filter (vec08u * dst, vec08u * src, int sstr, int bw, int bh)

5x5 gaussian filter.

Calculate the output of 5x5 gaussian filter.

dst	- [Output] Destination block pointer
src	- [Input] Source block pointer
sstr	- [Input] Source block stride
bw	- [Input] Block width
bh	- [Input] Block height

5.32 Image gradient 147

5.32 Image gradient

5.32.1 Detailed Description

Image gradient.

Collaboration diagram for Image gradient:



Functions

- KERNEL_INFO apu_gradient_out08s (" apu_gradient_out08s ", 3, __port(__index(0), __identifier("IN← PUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUT_GX"), __attributes(ACF_ATTR_VEC_← OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ← index(2), __identifier("OUT_GY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_gr_abs ("apu_gr_abs ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("OUT_ABSSUM"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void apuGradient (vec16s *apcSobelX, vec16s *apcSobelY, const vec08u *apInput, int aBlockWidth, int a
 —
 BlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

void apuGradient_out08s (vec08s *apcSobelX, vec08s *apcSobelY, const vec08u *apInput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

void apuGradientAbs (vec08s *apcSobelX, vec08s *apcSobelY, vec08u *apcAbsSum, const vec08u *apc
Input, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient ==> GradX, GradY, |GradX| + |GradY|.

Image gradient absulute sum ==> |GradX| + |GradY|.

5.32.2 Function Documentation

Image gradient kernel metadata.

Parameters

GRADIENT_A↔	Define for Kernel name
BS_KN	
2	Number of ports
Port GRADIEN↔	Define for name of input image (unsigned 8bit)
T_KN_IN	
Port	Define for name of sum of absolute values of grad X/Y components output image (unsigned
GRADIENT_K↔	16bit)
N_ABSGRAD	

```
5.32.2.2 KERNEL_INFO apu_gradient ( " apu_gradient " , 3 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUT_GX"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUT_GY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image gradient kernel metadata.

Parameters

GRADIENT_KN	Define for Kernel name
3	Number of ports
Port GRADIEN↔	Define for name of input image (unsigned 8bit)
T_KN_IN	
Port GRADIEN↔	Define for name of Gradient X component output image (signed 16bit)
T_KN_OUTX	
Port GRADIEN↔	Define for name of Gradient Y component output image (signed 16bit)
T_KN_OUTY	

```
5.32.2.3 KERNEL_INFO apu_gradient_abs ( "apu_gradient_abs " , 4 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUT_GX"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("OUT_GY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("OUT_ABSSUM"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image gradient kernel metadata.

GRADIENT A.	Define for Kernel name
GUADIENI_A⇔	Define for Kernel name
BS KN	

5.32 Image gradient 149

4	Number of ports
Port GRADIEN←	Define for name of input image (unsigned 8bit)
T_KN_IN	
Port GRADIEN←	Define for name of Gradient X component output image (signed 16bit)
T_KN_OUTX	
Port GRADIEN↔	Define for name of Gradient Y component output image (signed 16bit)
T_KN_OUTY	
Port	Define for name of sum of absolute values of grad X/Y components output image (unsigned
GRADIENT_K↔	16bit)
N_ABSGRAD	

5.32.2.4 void apuGradAbs (vec08u * apcAbsSum, const vec08u * apInput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient absulute sum ==> |GradX| + |GradY|.

Computes image gradient (horizontal and vertical Sobel filter). Returns sum of their absolute values

Parameters

apcAbsSum	- [Output] Sum of absolute values of GradX/Y destination block pointer (unsigned 16bit)
apInput	- [Input] Source block pointer (unsigned 8bit)
aBlockWidth	- [Input] Block width
aBlockHeight	- [Input] Block height
aStride	- [Input] Source block stride

5.32.2.5 void apuGradient (vec16s * apcSobelX, vec16s * apcSobelY, const vec08u * apInput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

Computes image gradient (horizontal and vertical Sobel filter). Returns Gradient X/Y

Parameters

apcSobelX	- [Output] Horizontal Sobel Filter destination block pointer (signed 16bit)
apcSobelY	- [Output] Vertical Sobel Filter destination block pointer (signed 16bit)
apInput	- [Input] Source block pointer (unsigned 8bit)
aBlockWidth	- [Input] Block width
aBlockHeight	- [Input] Block height
aStride	- [Input] Source block stride

5.32.2.6 void apuGradient_out08s (vec08s * apcSobelX, vec08s * apcSobelY, const vec08u * apInput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

Computes image gradient (horizontal and vertical Sobel filter). Returns Gradient X/Y

apcSobelX	- [Output] Horizontal Sobel Filter destination block pointer (signed 08bit)
apcSobelY	- [Output] Vertical Sobel Filter destination block pointer (signed 08bit)
aplnput	- [Input] Source block pointer (unsigned 8bit)

aBlockWidth	- [Input] Block width
aBlockHeight	- [Input] Block height
aStride	- [Input] Source block stride

5.32.2.7 void apuGradientAbs (vec08s * apcSobelX, vec08s * apcSobelY, vec08u * apcAbsSum, const vec08u * aplnput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient ==> GradX, GradY, |GradX|+|GradY|.

Computes image gradient (horizontal and vertical Sobel filter). Returns Gradient X/Y and sum of their absolute values

apcSobelX	- [Output] Horizontal Sobel Filter destination block pointer (signed 8bit)
apcSobelY	- [Output] Vertical Sobel Filter destination block pointer (signed 8bit)
apcSobelY	- [Output] Sum of absolute values of GradX/Y destination block pointer (unsigned 16bit)
apInput	- [Input] Source block pointer (unsigned 8bit)
aBlockWidth	- [Input] Block width
aBlockHeight	- [Input] Block height
aStride	- [Input] Source block stride

5.33 Non-maximum suppression

5.33.1 Detailed Description

Non-maximum suppression.

Collaboration diagram for Non-maximum suppression:



Functions

- KERNEL_INFO apu_nms (" apu_nms ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_
 ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
 e0 data type(d08u), e0 size(1, 1), ek size(1, 1)))
- KERNEL_INFO apu_nms16 ("apu_nms16", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- void apu_nms_impl (const vec08u *apcln, vec08u *apOut, int alnStride, int aOutStride, int aTileWidth, int aTileHeight)

Non-maximum suppression.

void apu_nms16 (const vec16u *apcln, vec16u *apOut, int alnStride, int aOutStride, int aTileWidth, int a
 —
 TileHeight)

Non-maximum suppression, 16-bit.

5.33.2 Function Documentation

```
5.33.2.1 KERNEL_INFO apu_nms ( "apu_nms " , 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Non-maximum suppression kernel metadata.

NMS_KN	Define for Kernel name
2	Number of ports
Port	Define for name of input image (unsigned 8bit)
NMS16_KN_IN	
Port NMS16_K←	Define for name of output image (unsigned 8bit)
N OUT	

5.33.2.2 void apu_nms16 (const vec16u * apcIn, vec16u * apOut, int alnStride, int aOutStride, int aTileWidth, int aTileHeight)

Non-maximum suppression, 16-bit.

Sets values which are not maximal in their 3x3 neighborhood (8 pixels) to 0. 16-bit version.

Parameters

apcIn	- [Input] Pointer to the source buffer
apOut	- [Output] Pointer to the destination buffer
alnStride	- [Input] Line stride of the source data
aOutStride	- [Input] Line stride of the destination data
aTileWidth	- [Input] Width of one data block
aTileHeight	- [Input] Height of one data block

```
5.33.2.3 KERNEL_INFO apu_nms16 ( "apu_nms16 ", 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

16-bit non-maximum suppression kernel metadata.

Parameters

NMS16_KN	Define for Kernel name
2	Number of ports
Port	Define for name of input image (unsigned 16bit)
NMS16_KN_IN	
Port NMS16_K↔	Define for name of output image (unsigned 16bit)
N_OUT	

5.33.2.4 void apu_nms_impl (const vec08u * apcIn, vec08u * apOut, int aInStride, int aOutStride, int aTileWidth, int aTileHeight)

Non-maximum suppression.

Sets values which are not maximal in their 3x3 neighborhood (8 pixels) to 0. 8-bit version

apcIn	- [Input] Pointer to the source buffer
apOut	- [Output] Pointer to the destination buffer
alnStride	- [Input] Line stride of the source data
aOutStride	- [Input] Line stride of the destination data
aTileWidth	- [Input] Width of one data block
aTileHeight	- [Input] Height of one data block

5.34 Row_filter 153

5.34 Row filter

5.34.1 Detailed Description

Filter an image column with a row_filter(i.e. Row_Vector * Matrix multiplication)

Collaboration diagram for Row filter:



Functions

- KERNEL_INFO row_filter (" row_filter ", 3, __port(__index(0), __identifier("SRC"), __attributes(ACF_ATT ← R_VEC_IN), __spatial_dep(0, 0, ROW_PADDING, ROW_PADDING), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STAT ← IC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_ROWS, 1)), __port(__index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void row_filter_impl (vec08u *dst, int dstStride, const vec08u *src, int srcStride, int rows, int cols, const unsigned char *filter, int filterSize, int filterQ)

Variables

- const int **FILTER_ROWS** = 5
- const int ROW PADDING = FILTER ROWS >> 1

5.34.2 Function Documentation

```
5.34.2.1 KERNEL_INFO row_filter ( " row_filter " , 3 , __port(__index(0), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, ROW_PADDING, ROW_PADDING), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_ROWS, 1)) , __port(__index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

ROW filter kernel metadata.

ROW_FILTER⇔	Define for Kernel name
_KN	
3	Number of ports
Port ROW_FIL←	Define for name of input image (unsigned 8bit)
TER KN SRC	

Port	Define for name of filter coefficients (unsigned 8bit)
ROW_FILTER↔	
_KN_COEFFS	
Port ROW_FIL↔	Define for name of output image (unsigned 8bit)
TER_KN_DST	

5.34.2.2 void row_filter_impl (vec08u * dst, int dstStride, const vec08u * src, int srcStride, int rows, int cols, const unsigned char * filter, int filterSize, int filterQ)

dst	Pointer to the destination image.
dstStride	Stride of the destination image.
src	Pointer to the source image.
srcStride	Stride of the source image.
rows	Rows of the source image, not including padding.
cols	Columns of the source image, not including padding.
filter	Pointer to the row filter coefficients. The coefficients are unsigned 8-bit fixed point numbers.
filterSize	Size of the column filter.
filterQ	Number of fractional bits for the filter coefficients.

5.35 Saturation 155

5.35 Saturation

5.35.1 Detailed Description

Saturation.

Collaboration diagram for Saturation:



Functions

- KERNEL_INFO apu_saturate_nonzero (" apu_saturate_nonzero ", 2, __port(_index(0), __identifier("INP← UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_binarize (" apu_binarize ", 2, __port(__index(0), __identifier("BIN_IN"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("BIN_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
 __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_mask (" apu_mask ", 5, __port(__index(0), __identifier("MASK_INFL"), __attributes(A \(\cup \) CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MASK_INX"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("MASK_INY"), \(\cup \) attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __\(\cup \) ek_size(1, 1)), __port(__index(3), __identifier("MASK_OUTX"), __attributes(ACF_ATTR_VEC_OUT), _\(\cup \) spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), _\(\cup \) identifier("MASK_OUTY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_\(\cup \) type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- void saturate_nonzero (vec08u *dst, vec08u *src, int dstr, int sstr, int bw, int bh)
 Non-zero pixel saturation.
- void binarize (vec08u *dst, vec32u *src, int, int, int bw, int bh)

Non-zero pixel binarization.

void mask (vec16s *dstX, vec16s *dstY, vec32u *srcFlags, vec16s *inX, vec16s *inY, int, int, int bw, int bh)
 Masking to zero with the srcFlags.

5.35.2 Function Documentation

5.35.2.1 KERNEL_INFO apu_binarize ("apu_binarize ", 2, __port(_index(0), __identifier("BIN_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("BIN_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Non-zero saturation kernel metadata.

Parameters

SAT_NONZER⇔	Define for Kernel name
O_KN	
2	Number of ports
Port SAT_NO⇔	Define for name of input image (unsigned 8bit)
NZERO_IN	
Port SAT_NO⇔	Define for name of output image (unsigned 8bit)
NZERO_OUT	

```
5.35.2.2 KERNEL_INFO apu_mask ( "apu_mask ", 5 , __port(_index(0), __identifier("MASK_INFL"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("MASK_INX"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("MASK_INY"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("MASK_OUTX"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(4), __identifier("MASK_OUTY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

Masking kernel metadata. Mask input to zero if flags are zero

Parameters

MASK_KN	Define for Kernel name
5	Number of ports
Port MASK_K_←	Define for name of masking flags input image (unsigned 32bit)
INFLAGS	
Port MASK_INX	Define for name of input X image to be masked(signed 16bit)
Port MASK_INY	Define for name of input Y image to be masked(signed 16bit)
Port	Define for name of output X image (signed 16bit)
MASK_OUTX	
Port	Define for name of output Y image (signed 16bit)
MASK_OUTY	

```
5.35.2.3 KERNEL_INFO apu_saturate_nonzero ( "apu_saturate_nonzero ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Non-zero saturation kernel metadata.

Parameters

SAT_NONZER↔	Define for Kernel name
O_KN	
2	Number of ports
Port SAT_NO↔	Define for name of input image (unsigned 8bit)
NZERO_IN	
Port SAT_NO↔	Define for name of output image (unsigned 8bit)
NZERO OUT	

5.35.2.4 void binarize (vec08u * dst, vec32u * src, int , int , int bw, int bh)

Non-zero pixel binarization.

Changes non-zero pixel values to maximal values.

5.35 Saturation 157

Parameters

dst	- [Output] Pointer to the destination buffer (unsigned 8bit)
src	- [Input] Pointer to the source buffer (unsigned 32bit)
dstr	- [Input] Line stride of the destination data
sstr	- [Input] Line stride of the source data
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block

5.35.2.5 void mask (vec16s * dstX, vec16s * dstY, vec32u * srcFlags, vec16s * inX, vec16s * inY, int , int bw, int bh)

Masking to zero with the srcFlags.

Masks the output pixel values to zero, if the srcFlags are zero. Useful to mask thresholded GradientX/Y

Parameters

dstX	- [Output] Pointer to the X destination buffer (signed 16bit)
dstY	- [Output] Pointer to the Y destination buffer (signed 16bit)
srcFlags-	[Input] Pointer to the mask source buffer (unsigned 32bit)
inX	- [Output] Pointer to the X source buffer to be masked (signed 16bit)
inY	- [Output] Pointer to the Y source buffer to be masked(signed 16bit)
dstr	- [Input] Line stride of the destination data
sstr	- [Input] Line stride of the source data
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block

5.35.2.6 void saturate_nonzero (vec08u * dst, vec08u * src, int dstr, int sstr, int bw, int bh)

Non-zero pixel saturation.

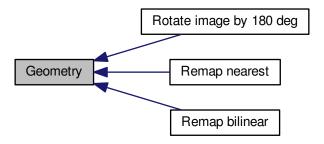
Changes non-zero pixel values to maximal values.

dst	- [Output] Pointer to the destination buffer (unsigned 8bit)
src	- [Input] Pointer to the source buffer (unsigned 8bit)
dstr	- [Input] Line stride of the destination data
sstr	- [Input] Line stride of the source data
bw	- [Input] Width of one data block
bh	- [Input] Height of one data block

5.36 Geometry

5.36.1 Detailed Description

Collaboration diagram for Geometry:



Modules

• Remap bilinear

Element-wise bilinear interpolation.

Remap nearest

Element-wise nearest neighbor interpolation.

• Rotate image by 180 deg

Image rotation.

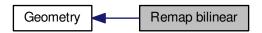
5.37 Remap bilinear 159

5.37 Remap bilinear

5.37.1 Detailed Description

Element-wise bilinear interpolation.

Collaboration diagram for Remap bilinear:



Functions

- void remap_bilinear_rgb_impl (vec32u *dst, vec32u *src, vec16u *offset, vec08u *delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

void remap_bilinear_grayscale_impl (vec08u *dst, vec08u *src, vec16u *offset, vec08u *delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

5.37.2 Function Documentation

5.37.2.1 KERNEL_INFO remap_bilinear_grayscale (" remap_bilinear_grayscale " , 4 , __port(__index(0), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(36, 12)) , __port(__index(2), __identifier("OFFSET"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("DELTA"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(2, 1), __ek_size(1, 1)))

GrayScale interpolation kernel metadata. Interpolates between neighboring pixels of a grayscale image. Outputs 8bit unsigned interpolation result.

Parameters

REMAP_BILIN↔	Define for Kernel name
_GRAY_KN	
4	Number of ports
Port REMAP_←	Define for name of interpolation result (unsigned 8bit).
BILIN_KN_DST	
Port REMAP_←	Define for name of first input image (unsigned 8bit)
BILIN_KN_SRC	
Port	Define for name of input offsets (unsigned 16bit)
REMAP_BILIN↔	
_KN_OFFS	
Port	Define for name of second input deltas (unsigned 8bit)
REMAP_BILIN↔	
_KN_DELTA	

5.37.2.2 void remap_bilinear_grayscale_impl (vec08u * dst, vec08u * src, vec16u * offset, vec08u * delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

remap_bilinear btw the grayscale pixels of an image

Parameters

dst	- [Output] unsigned 8bit destination block pointer
srcImage0-	[Input] unsigned 8bit source block pointer of img 0
offset	- [Input] unsigned 16bit offsets inside source block pointer of img 0
delta	- [Input] unsigned 8bit deltas inside source block pointer of img 0
sstride	- [Input] Source block width (in elements not bytes) including padding
dstride	- [Input] Destination block width (in elements not bytes) including padding
bw	- [Input] Block width
bh	- [Input] Block height

```
5.37.2.3 KERNEL_INFO remap_bilinear_rgb ( " remap_bilinear_rgb ", 4 , __port(_index(0), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(8, 9)) , __port(_index(2), __identifier("OFFSET"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("DELTA"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(2, 1), __ek_size(1, 1)) )
```

RGB interpolation kernel metadata. Interpolates between neighboring pixels of a RGB image. Outputs 8bit unsigned interpolation result.

REMAP_BILIN↔	Define for Kernel name
_GRAY_KN	
4	Number of ports
Port REMAP_←	Define for name of interpolation result (unsigned 32bit).
BILIN_KN_DST	

5.37 Remap bilinear 161

Port REMAP_←	Define for name of first input image (unsigned 32bit)
BILIN_KN_SRC	
Port	Define for name of input offsets (unsigned 16bit)
REMAP_BILIN↔	
_KN_OFFS	
Port	Define for name of second input deltas (unsigned 8bit)
REMAP_BILIN↔	
_KN_DELTA	

5.37.2.4 void remap_bilinear_rgb_impl (vec32u * dst, vec32u * src, vec16u * offset, vec08u * delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

remap_bilinear btw the rgb pixels of an image

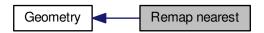
dst	- [Output] unsigned 32bit destination block pointer
srcImage0-	[Input] unsigned 32bit source block pointer of img 0
offset	- [Input] unsigned 16bit offsets inside source block pointer of img 0
delta	- [Input] unsigned 8bit deltas inside source block pointer of img 0
sstride	- [Input] Source block width (in elements not bytes) including padding
dstride	- [Input] Destination block width (in elements not bytes) including padding
bw	- [Input] Block width
bh	- [Input] Block height

5.38 Remap nearest

5.38.1 Detailed Description

Element-wise nearest neighbor interpolation.

Collaboration diagram for Remap nearest:



Functions

- KERNEL_INFO remap_nearest_grayscale (" remap_nearest_grayscale ", 3, __port(_index(0), __
 identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN),
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __
 identifier("OFFSET"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u),
 __e0_size(1, 1), __ek_size(1, 1)))
- void remap_nearest_grayscale_impl (vec08u *dst, vec08u *src, vec16u *offset, int sstride, int dstride, int bw, int bh)

Elementwise nearest neighbor interpolation.

5.38.2 Function Documentation

```
5.38.2.1 KERNEL_INFO remap_nearest_grayscale ( "remap_nearest_grayscale ", 3, __port(__index(0), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OFFSET"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

GrayScale interpolation kernel metadata. Interpolates between neighboring pixels of a grayscale image. Outputs 8bit unsigned interpolation result.

REMAP_NEA⇔	Define for Kernel name
REST_GRAY_←	
KN	
4	Number of ports
Port	Define for name of interpolation result (unsigned 8bit).
REMAP_NEA↔	
REST_KN_DST	

5.38 Remap nearest 163

Port	Define for name of first input image (unsigned 8bit)
REMAP_NEA⇔	
REST_KN_SRC	
Port REMAP_←	Define for name of input offsets (unsigned 16bit)
NEAREST_KN↔	
_OFFS	

5.38.2.2 void remap_nearest_grayscale_impl (vec08u * dst, vec08u * src, vec16u * offset, int sstride, int dstride, int bw, int bh)

Elementwise nearest neighbor interpolation.

remap_nearest btw the grayscale pixels of an image

dst	- [Output] unsigned 8bit destination block pointer
srcImage0-	[Input] unsigned 8bit source block pointer of img 0
offset	- [Input] unsigned 16bit offsets inside source block pointer of img 0
sstride	- [Input] Source block width (in elements not bytes) including padding
dstride	- [Input] Destination block width (in elements not bytes) including padding
bw	- [Input] Block width
bh	- [Input] Block height

5.39 Rotate image by 180 deg

5.39.1 Detailed Description

Image rotation.

Collaboration diagram for Rotate image by 180 deg:



Macros

- #define ROTATE_K apu_rotate_180
- #define ROTATE_KN XSTR(ROTATE_K)
- #define ROTATE KN IN "INPUT"
- #define ROTATE_KN_OUT "OUTPUT"

Functions

- KERNEL_INFO apu_rotate_180 (" apu_rotate_180 ", 2, __port(__index(0), __identifier("INPUT"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek
 _size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- extKernelInfoDecl (ROTATE_180_K)
- void apu_rotate_180 (kernel_io_desc IIn, kernel_io_desc IOut)
- extKernelInfoDecl (apu_rotate_180)
- void rotate_180 (vec08u *dst, vec08u *src, int bw, int bh, int sstr)
 180-degree rotation.

5.39.2 Function Documentation

180-degree rotation kernel metadata.

ROTATE_KN	Define for Kernel name
2	Number of ports
Port ROTATE_←	Define for name of input image (unsigned 8bit)
KN_IN	

Port ROTATE_←	Define for name of output image (unsigned 8bit)
KN_OUT	

5.39.2.2 void rotate_180 (vec08u * dst, vec08u * src, int bw, int bh, int sstr)

180-degree rotation.

Rotates the image 180 degrees.

dst	- [Output] pointer to the destination buffer
src	- [Input] pointer to the source buffer
bw	- [Input] width of one data block
bh	- [Input] height of one data block
sstr	- [Input] source stride

5.40 Morphology

5.40.1 Detailed Description

Collaboration diagram for Morphology:



Modules

Dilation

Image dilation.

5.41 Dilation 167

5.41 Dilation

5.41.1 Detailed Description

Image dilation.

Collaboration diagram for Dilation:



Functions

- void dilate_diamond (vec08u *dst, vec08u *src, int dstr, int sstr, int bw, int bh)
 5x5 diamond dilation.

5.41.2 Function Documentation

```
5.41.2.1 KERNEL_INFO apu_dilate_diamond ( " apu_dilate_diamond " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Diamond dilation kernel metadata.

Parameters

<i>DILATE_DIAM</i> ←	Define for Kernel name
OND_KN	
2	Number of ports
Port	Define for name of input image (unsigned 8bit)
<i>DILATE_DIAM</i> ←	
OND_KN_IN	
Port	Define for name of output image (unsigned 8bit)
<i>DILATE_DIAM</i> ←	
OND_KN_OUT	

5.41.2.2 void dilate_diamond (vec08u * dst, vec08u * src, int dstr, int sstr, int bw, int bh)

5x5 diamond dilation.

Dilates the image using 5x5 diamond structure element.

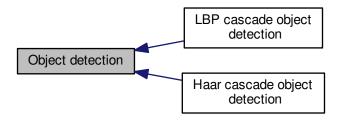
dst	- [Output] pointer to the destination buffer
src	- [Input] pointer to the source buffer
dstr	- [Input] line stride of the destination data
sstr	- [Input] line stride of the source data
bw	- [Input] width of one data block
bh	- [Input] height of one data block

5.42 Object detection 169

5.42 Object detection

5.42.1 Detailed Description

Collaboration diagram for Object detection:



Modules

- Haar cascade object detection
 Haar cascade object detection.
- LBP cascade object detection

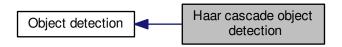
 LBP cascade object detection.

5.43 Haar cascade object detection

5.43.1 Detailed Description

Haar cascade object detection.

Collaboration diagram for Haar cascade object detection:



Classes

- struct APEX_HaarCascadeFeature
- struct APEX HaarCascadeStage

Typedefs

- typedef vec16u FEATURE FIXED_POINT_TYPE
- typedef vec16u STAGE_FIXED_POINT_TYPE

Functions

- KERNEL_INFO apu_haar_cascade (" apu_haar_cascade ", 11, __port(__index(0), __identifier("INPUT_I← NTEGRAL_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), _e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_INTEGRAL_IMAGE_SQUARE↔ $D"),\ \underline{\quad} attributes(ACF_ATTR_VEC_IN),\ \underline{\quad} spatial_dep(1,\ 0,\ 1,\ 0),\ \underline{\quad} e0_data_type(d32u),\ \underline{\quad} e0_size(1,\ 1),\ \underline{\quad} spatial_dep(1,\ 0,\ 1,\ 0),\ \underline{\quad} e0_data_type(d32u),\ \underline{\quad} e0_size(1,\ 1),\ \underline{\quad} e0_size($ _ek_size(1, 1)), __port(__index(2), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_SCL_OUT_ST↩ $ATIC_FIXED), \underline{\hspace{0.5cm}} spatial_dep(0, 0, 0, 0), \underline{\hspace{0.5cm}} e0_data_type(d16u), \underline{\hspace{0.5cm}} e0_size(1, 1), \underline{\hspace{0.5cm}} ek_size(1, 1)), \underline{\hspace{0.5cm}} port(\leftarrow$ _index(3), __identifier("WINDOW_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __← spatial dep(0, 0, 0, 0), e0 data type(d32u), e0 size(1, 1), ek size(6+1, 32)), port $(index(4), \leftarrow$ spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(6+1, 32)), __port(__index(5), $\underline{\quad } identifier("INPUT_CASCADE_SIZES"), \ \underline{\quad } attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), \ \underline{\quad } spatial_{\ } \hookrightarrow CASCADE_SIZES"$ dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6), __identifier("I← NPUT_CASCADE_FEATURES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(19800, 1)), __port(__index(7), __identifier("IN← PUT CASCADE STAGES"), attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, 0, 0, 0), L_SHIFTS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ $type(d08u), \underline{\hspace{0.5cm}} e0_size(1, 1), \underline{\hspace{0.5cm}} ek_size(32, 1)), \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(9), \underline{\hspace{0.5cm}} identifier("INPUT_PIXEL_OFFSETS"), \underline{\hspace{0.5cm}} identifier("INPUT_PIXEL_OFFSETS"), \underline{\hspace{0.5cm}} identifier(\underline{\hspace{0.5cm}} index(9), \underline{\hspace{0.5cm}} inde$ _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), _ e0_size(1, 1), __ek_size(32, 1)), __port(__index(10), __identifier("OUTPUT"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void haar_cascade (vec08u *apOut, vec32u *apInII, vec32u *apInII2, int aOutStride, int aInStride, int a
 — TileWidth, int aTileHeight, int16u aLineIndex, vec32u *apWindowBuffer, vec32u *apWindowBuffer2, const
 APEX_HaarCascadeFeature *apcFeatures, int aStageCount, const APEX_HaarCascadeStage *apcStages,
 const int08u *apcXshifts, const int08u *apcXoffsets)

Haar object detection.

Variables

- const int **FEATURE_FRACTIONAL_BITS** = 13
- const int FEATURE_FIXED_POINT_MULTIPLIER = (1 << FEATURE_FRACTIONAL_BITS)
- const int STAGE_FRACTIONAL_BITS = 10
- const int STAGE FIXED POINT MULTIPLIER = (1 << STAGE FRACTIONAL BITS)
- const int32u featureFixedCoefficientSqrt = 91
- const int32s invWindowAreaScalar = 25

5.43.2 Function Documentation

```
5.43.2.1 KERNEL_INFO apu_haar_cascade ( " apu_haar_cascade " , 11 , __port(_index(0), __identifier("INPU -
        T_INTEGRAL_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u),
         __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("INPUT_INTEGRAL_IMAGE_SQUARED"),
         __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) ,
         __port(_index(2), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0,
        0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("WINDOW_BUFFER"),
         _attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u),
         __e0_size(1, 1), __ek_size(6+1, 32)) , __port(__index(4), __identifier("WINDOW_BUFFER_SQUARED"),
         _attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u),
         e0 size(1, 1), ek size(6+1, 32)), port( index(5), identifier("INPUT CASCADE SIZES"),
         attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d16u), e0 size(1, 1),
          SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(19800, 1))
         , __port(_index(7), __identifier("INPUT_CASCADE_STAGES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED),
         __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(70, 1)) , __port(__index(8),
         __identifier("INPUT_PIXEL_SHIFTS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0),
         _e0_data_type(d08u), _e0_size(1, 1), _ek_size(32, 1)) , _port(_index(9), _identifier("INPUT_PIXEL_OFFSETS"),
         attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d08u), e0 size(1, 1),
          _ek_size(32, 1)), __port(__index(10), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0,
         0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Haar cascade object detection kernel metadata.

apu_haar_⇔	Kernel name
cascade	
11	Number of ports
Port HAAR_C↔	Define for name of input integral image (unsigned 32bit)
ASC_IN_INTE↔	
GR_IMG	
Port HAAR_C↔	Define for name of input integral image of squared values (unsigned 32bit)
ASC_IN_INTE↔	
GR_IMG_SQR	
Port HAAR_C↔	Define for name of line index buffer (unsigned 16bit)
ASC_LINE_IDX	
Port	Define for name of integral image window buffer (unsigned 32bit)
HAAR_CASC_←	
WND BUF	

Port HAAR_C↔	Define for name of integral image of squared values window buffer (unsigned 32bit)
ASC_WND_B⇔	
UF_SQR	
Port	Define for name of number of features and stages (2 16-bit values) (unsigned 16bit)
HAAR_CASC_←	
IN_CASC_SZ	
Port HAAR_C↔	Define for name of array of Haar-like features (unsigned 8bit)
ASC_IN_CAS⇔	
C_FEAT	
Port HAAR_C↔	Define for name of array of cascade stages (unsigned 8bit)
ASC_IN_CAS←	
C_STAGES	
Port	Define for name of LUT containing tile shifts needed for horizontal pixel offsets (unsigned
HAAR_CASC_←	8bit)
IN_PIX_SHFT	
Port HAAR_C↔	Define for name of LUT containing in-tile offsets needed for horizontal pixel offsets (unsigned
ASC_IN_PIX_←	8bit)
OFFSETS	
Port HAAR_C↔	Define for name of output image (unsigned 8bit)
ASC_OUT	

5.43.2.2 void haar_cascade (vec08u * apOut, vec32u * apInII, vec32u * apInII2, int aOutStride, int aInStride, int aTileWidth, int aTileHeight, int16u aLineIndex, vec32u * apWindowBuffer, vec32u * apWindowBuffer2, const APEX_HaarCascadeFeature * apcFeatures, int aStageCount, const APEX_HaarCascadeStage * apcStages, const int08u * apcXshifts, const int08u * apcXshifts)

Haar object detection.

Detects objects using Haar-like feature cascades. The algorithm searches for 20x20-pixel objects using a Haar-like classifier provided by the user. For each input pixel, it outputs 255 if the pixel is a lower left corner of an object and 0 otherwise. The classifier consists of a number of stages, each stage contains multiple Haar-like features. The window has to pass every stage in order to be classified as containing an object.

See http://docs.opencv.org/trunk/doc/py_tutorials/py_objdetect/py_face_← detection/py_face_detection.html

The format of the classifier stage is:

```
struct APEX_HaarCascadeStage
{
  int08u featureCount;
  int16u thresholdFixed;
}.
```

Where feature Count is a number of Haar-like features included in the stage.

ThresholdFixed is a fixed point threshold (10 fractional bits) which has to be exceeded by the sum of the feature test results in order for the stage to pass.

A Haar-like feature is a rectangle subdivided into weighted sub-rectangles. The pixels inside the feature region are weighted according to the weight of the sub-rectangle they belong to and then they are summed together. The resulting sum is then compared with the feature's threshold and the resulting boolean tells whether the feature passed or not.

```
struct APEX_HaarCascadeStage
{
  int16u threshold;
  int16u leftVal, rightVal;
  int16u x5y5w5_1;
  int08u h5type3;
};
```

Where threshold is a fixed-point number (13 fractional bits) which has to be exceeded by the weighted sum of the feature rectangles in order for the feature to pass.

leftVal, rightVal are fixed-point values (10 fractional bits), leftVal is added to the stage sum if the feature doesn't pass, righVal is added to the stage sum if the feature passes.

x5y5w5_1 contains 3 values in its bits, starting from the most significant bit: upper left corner feature position x component in the first 5 bits, the y component in the next bits and the width of the feature in the next 5 bits. The least significant bit is not used.

h5type3 contains 2 values in its bits, starting from the most significant bit: feature height in the first bit, feature type in the remaining 3 bits.

There are 8 possible types of feature:

type 0 ... horizontal edge, upper half positive

type 1 ... horizontal edge, lower half positive

type 2 ... vertical edge, left half positive

type 3 ... vertical edge, right half positive

type 4 ... line, horizontal

type 5 ... line, vertical

type 6 ... 4 rectangles, upper left and lower right rectangles positive ()

type 7 ... 4 rectangles, lower left and upper right rectangles positive (/)

The kernel expects an integral image and an integral image of squared values. The implementation retains a rolling window for the integral images, therefore sufficiently sized buffers have to be supplied.

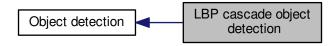
- [Output] pointer to the destination buffer
- [Input] pointer to the source integral image buffer
- [Input] pointer to the source integral image of squared values buffer
- [Input] line stride of the destination data
- [Input] line stride of the source data
- [Input] width of one data tile
- [Input] height of one data tile
- [Input] index of currently processed line
- [Buffer] pointer to the input integral image window buffer
- [Buffer] pointer to the input integral image of squared values window buffer
- [Input] pointer to the Haar-like feature array
- [Input] number of Haar-like feature cascade stages
- [Input] pointer to the Haar-like feature cascade stages
- [Input] pointer to LUT containing tile shifts needed for horizontal pixel offsets
- [Input] pointer to LUT containing in-tile offsets needed for horizontal pixel offsets

5.44 LBP cascade object detection

5.44.1 Detailed Description

LBP cascade object detection.

Collaboration diagram for LBP cascade object detection:



Classes

- struct APEX_lbpFeature
- struct APEX lbpStage

Typedefs

- typedef vec32s STAGE FIXED POINT TYPE
- typedef int32s STAGE_FIXED_POINT_TYPE_SCALAR

Functions

- KERNEL_INFO apu_lbp_cascade (" apu_lbp_cascade ", 9, __port(__index(0), __identifier("INPUT_INTE--GRAL_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), __ $e0_size(1,1), __ek_size(1,1)), __port(__index(1), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_S \leftarrow ACF_ATTR_S)$ CL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("WINDOW_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_← FIXED), spatial dep(0, 0, 0, 0), e0 data type(d32u), e0 size(1, 1), ek size(20+1, 32)), port $(\leftarrow$ _index(3), __identifier("INPUT_CASCADE_SIZES_AND_SKIP"), __attributes(ACF_ATTR_SCL_IN_STA↔ TIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(3, 1)), __port(← _index(4), __identifier("INPUT_CASCADE_FEATURES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FI↔ $XED), \underline{\hspace{0.5cm}} spatial_dep(0, \ 0, \ 0, \ 0), \ \underline{\hspace{0.5cm}} e0_data_type(d08u), \ \underline{\hspace{0.5cm}} e0_size(1, \ 1), \ \underline{\hspace{0.5cm}} ek_size(10000, \ 1)), \ \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} \leftarrow b)$ _index(5), __identifier("INPUT_CASCADE_STAGES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), spatial dep(0, 0, 0, 0), e0 data type(d08u), e0 size(1, 1), ek size(200, 1)), port $(index(6), \leftarrow$ identifier("INPUT PIXEL SHIFTS"), attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, (0, 0, 0), e0 data type(d08u), e0 size(1, 1), ek size(64, 1)), port(index(7), identifier("INP \leftarrow UT_PIXEL_OFFSETS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), _ <-_e0_data_type(d08u), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(8), __identifier("OUTPUT"), __ attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __e ek_size(1, 1)))
- void lbp_cascade (vec08u *apOut, vec32u *apInII, int aOutStride, int aInStride, int aTileWidth, int aTileHeight, int16u aLineIndex, vec32u *apWindowBuffer, const APEX_lbpFeature *apcFeatures, int aStageCount, const APEX_lbpStage *apcStages, const int08u *apcXshifts, const int08u *apcXoffsets, int skipOdd)

Local Binary Pattern object detection.

Variables

- const int STAGE FRACTIONAL BITS = 28
- const int **STAGE_FIXED_POINT_MULTIPLIER** = (1 << STAGE_FRACTIONAL_BITS)

5.44.2 Function Documentation

5.44.2.1 KERNEL_INFO apu_lbp_cascade (" apu_lbp_cascade " , 9 , __port(__index(0), __identifier("INPUT_INTEGRAL_IMA --GE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, $0,\,0,\,0),\,_e0_data_type(d16u),\,_e0_size(1,\,1),\,_ek_size(1,\,1))\,\,,\,_port(_index(2),\,_identifier("WINDOW_BUFFER"),$ __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), _e0_size(1, 1), _ek_size(20+1, 32)) , _port(_index(3), _identifier("INPUT_CASCADE_SIZES_AND_SKIP"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), _ek_size(3, 1)) , __port(__index(4), __identifier("INPUT_CASCADE_FEATURES"), __attributes(ACF_ATTR_-SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(10000, 1)) , __port(_index(5), __identifier("INPUT_CASCADE_STAGES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(200, 1)) , __port(__index(6), __identifier("INPUT_PIXEL_SHIFTS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), _e0_size(1, 1), _ek_size(64, 1)) , _port(_index(7), _identifier("INPUT_PIXEL_OFFSETS"), _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(64, 1)) , __port(__index(8), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

LBP cascade object detection kernel metadata.

LBP_CASC_KN	Define for Kernel name
9	Number of ports
Port	Define for name of input integral image (unsigned 32bit)
LBP_CASC_I⊷	
N_INTEGR_IMG	
Port LBP_CAS←	Define for name of line index buffer (unsigned 16bit)
C_LINE_IDX	
Port LBP_CAS↔	Define for name of integral image window buffer (unsigned 32bit)
C_WND_BUF	
Port LBP_CAS←	Define for name of number of features and stages + whetherthe odd rows and columns should
C_IN_CASC_←	be skipped (3 16-bit values) (unsigned 16bit)
SZ_AND_SKIP	
Port	Define for name of array of Haar-like features (unsigned 8bit)
LBP_CASC_I⊷	
N_CASC_FEAT	
Port LBP_CAS←	Define for name of array of cascade stages (unsigned 8bit)
C_IN_CASC_←	
STAGES	
Port	Define for name of LUT containing tile shifts needed for horizontal pixel offsets (unsigned
LBP_CASC_I⊷	8bit)
N_PIX_SHFT	
Port	Define for name of LUT containing in-tile offsets needed for horizontal pixel offsets (unsigned
LBP_CASC_I⇔	8bit)
N_PIX_OFFS	

Port LBP_CAS↔	Define for name of output image (unsigned 8bit)
C_OUT	

5.44.2.2 void lbp_cascade (vec08u * apOut, vec32u * apInII, int aOutStride, int aInStride, int aTileWidth, int aTileHeight, int16u aLineIndex, vec32u * apWindowBuffer, const APEX_lbpFeature * apcFeatures, int aStageCount, const APEX_lbpStage * apcStages, const int08u * apcXshifts, const int08u * apcXoffsets, int skipOdd)

Local Binary Pattern object detection.

Detects objects using local binary pattern feature cascades.

The algorithm searches for 24x24-pixel objects using a local binary pattern (LBP) classifier provided by the user. For each input pixel, it outputs 255 if the pixel is a lower left corner of an object and 0 otherwise. The classifier consists of a number of stages, each stage contains multiple LBP features. The window has to pass every stage in order to be classified as containing an object.

See Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang and Stan Z. Li. Learning Multi-scale Block Local Binary Patterns for Face Recognition

```
http://www.nlpr.labs.gov.cn/2007papers/gjhy/gh98.pdf
```

The format of the classifier stage is:

```
struct APEX_lbpStage
{
   uint8_t featureCount;
   int32s threshold;
};
```

Where featureCount is a number of LBP features included in the stage.

threshold is a fixed point threshold (28 fractional bits) which has to be exceeded by the sum of the feature test results in order for the stage to pass.

An LBP feature is a rectangle subdivided into a uniform 3x3 grid. Pixel values in each of the 9 sub-areas are summed. The sums of the outer areas are compared to the sum of the central area resulting in a 8-bit code (starting in the upper left area and going clockwise). This code is then used as an index to 256-bit table and the resulting bit tells whether the feature passed or not.

```
struct APEX_lbpFeature
{
  int32_t values[8];
  int32s leafValuesFixed[2];
  uint8_t x, y, width, height;
```

values[8] is a 256-bit classifying table used to determine whether the LBP feature with a given code should pass. Where threshold is a fixed-point number (13 fractional bits) which has to be exceeded by the weighted sum of the feature rectangles in order for the feature to pass.

leafValuesFixed are fixed-point values (28 fractional bits), leafValuesFixed[0] is added to the stage sum if the feature passes, leafValuesFixed[1] is added to the stage sum if the feature doesn't pass.

x, y define the position of the feature's upper left corner width, height define the size of the feature

The kernel expects an integral image. The implementation retains a rolling window for the integral image, therefore sufficiently sized buffer has to be supplied.

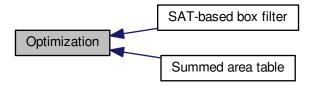
```
apOut - [Output] pointer to the destination buffer
```

aplnll	- [Input] pointer to the source integral image buffer
aOutStride	- [Input] line stride of the destination data
alnStride	- [Input] line stride of the source data
aTileWidth	- [Input] width of one data tile
aTileHeight	- [Input] height of one data tile
aLineIndex	- [Input] index of currently processed line
apWindowBuffer	- [Buffer] pointer to the input integral image window buffer
apcFeatures	- [Input] pointer to the LBP feature array
aStageCount	- [Input] number of LBP feature cascade stages
apcStages	- [Input] pointer to the LBP feature cascade stages
apcXshifts	- [Input] pointer to LUT containing tile shifts needed for horizontal pixel offsets
apcXoffsets	- [Input] pointer to LUT containing in-tile offsets needed for horizontal pixel offsets
skipOdd	- [Input] 1 if even columns and rows should be skipped, 0 otherwise

5.45 Optimization

5.45.1 Detailed Description

Collaboration diagram for Optimization:



Modules

• Summed area table

Summed area table (integral image) computation.

· SAT-based box filter

SAT-based box filter.

5.46 Summed area table 179

5.46 Summed area table

5.46.1 Detailed Description

Summed area table (integral image) computation.

Collaboration diagram for Summed area table:



Functions

- KERNEL_INFO apu_sat (" apu_sat ", 3, __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR \cdot VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_complex(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_complex_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_ROW"), __complex_dattributes(ACF_ATTR_VEC_OUT_STATIC), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- void sat32 (vec32u *apDest, vec32u *apPrevRow, const vec08u *apcSrc, int aSourceStride, int a
 — DestinationStride, int aBlockWidth, int aBlockHeight, int08u aFirstTile)

Summed area table.

5.46.2 Function Documentation

5.46.2.1 KERNEL_INFO apu_sat ("apu_sat ", 3, __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT \rightarrow PUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUTPUT_ROW"), __attributes(ACF_ATTR_VEC_OUT_STATIC), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

Summed area table kernel metadata.

Parameters

SAT_KN	Define for Kernel name
3	Number of ports
Port	Define for name of input image (unsigned 8bit)
SAT_KN_IN	
Port	Define for name of output image (unsigned 32bit)
SAT_KN_OUT	
Port SAT_KN_←	Define for name of last row of previous tile buffer (unsigned 32bit)
OUT_ROW	

5.46.2.2 void sat32 (vec32u * apDest, vec32u * apPrevRow, const vec08u * apcSrc, int aSourceStride, int aBlockWidth, int aBlockHeight, int08u aFirstTile)

Summed area table.

Computes summed area table (integral image).

out(i,j) = sum(in(k,l)) where k,l := (0,0) to (i,j) inclusive

apDest	- [Output] 32bit the integral image of the input
apPrevRow	- [Output] 32bit the last integral row in the tile.
apcSrc	- [Input] 8bit source block pointer
aSourceStride	- [Input] Source block width (in bytes) including padding
aDestination⊷	[Input] Destination block width (in bytes) including padding
Stride-	
aBlockWidth	- [Input] Width of one data tile
aBlockHeight	- [Input] Height of one data tile
aFirstTile	- [Input]Boolean. True, if the first tile is computed.

5.47 SAT-based box filter 181

5.47 SAT-based box filter

5.47.1 Detailed Description

SAT-based box filter.

Collaboration diagram for SAT-based box filter:



Functions

- KERNEL_INFO apu_sat_box_filter (" apu_sat_box_filter ", 2, __port(__index(0), __identifier("INPUT"), _
 _ attributes(ACF_ATTR_VEC_IN), __spatial_dep(BOX_SIZE+1, BOX_SIZE, BOX_SIZE+1, BOX_SIZE),
 — e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)))
- void sat_box_filter_impl (vec08u *apDest, const vec32u *apcSrc, int aBlockWidth, int aBlockHeight, int a
 — SourceStride, int aDestStride)

Sum of values over one patch the input image is a SAT image (i.e. integral computed with the sat32() function.

Variables

- · const int BOX_SIZE
- · const int BOX AREA
- · const int BOX SIZE
- · const int BOX_AREA

5.47.2 Function Documentation

```
5.47.2.1 KERNEL_INFO apu_sat_box_filter ( " apu_sat_box_filter " , 2 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(BOX_SIZE+1, BOX_SIZE, BOX_SIZE+1, BOX_SIZE), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Box filter using summed area table kernel metadata.

SAT_BOX_FIL↔	Define for Kernel name
TER_KN	
2	Number of ports
Port SAT_BOX⇔	Define for name of input summed area table image (unsigned 32bit)
_FILTER_IN	

Port SAT_BOX⇔	Define for name of output image (unsigned 8bit)
_FILTER_OUT	

5.47.2.2 void sat_box_filter_impl (vec08u * apDest, const vec32u * apcSrc, int aBlockWidth, int aBlockHeight, int aSourceStride, int aDestStride)

Sum of values over one patch the input image is a SAT image (i.e. integral computed with the sat32() function.

Applies a box filter (== sum over that patch) to the image using its summed area table (integral image).

 $out(i,j) = in(i - box_size - 1, j - box_size - 1) + in(i + box_size, j + box_size) - in(i - box_size - 1, j + box_size) - in(i + box_size, j - box_size - 1)$

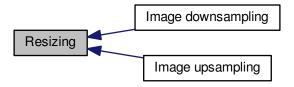
apDest	- [Output] 8bit destination image containing sum of values over one patch
apcSrc	- [Input] 32bit source block pointer
aBlockWidth	- [Input] Block width
aBlockHeight	- [Input] Block height
aSourceStride	- [Input] Source block width (in bytes) including padding
aDestination⇔	[Input] Destination block width (in bytes) including padding
Stride-	

5.48 Resizing 183

5.48 Resizing

5.48.1 Detailed Description

Collaboration diagram for Resizing:



Modules

- Image downsampling

 Image downsampling.
- Image upsampling

 Image upsampling.

5.49 Image downsampling

5.49.1 Detailed Description

Image downsampling.

Collaboration diagram for Image downsampling:



Functions

- KERNEL_INFO apu_downsample (" apu_downsample ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(2, 2)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_downsample_16u (" apu_downsample_16u ", 2, __port(__index(0), __identifier("INP←UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_downsample_gauss (" apu_downsample_gauss ", 2, __port(__index(0), __identifier("I← NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_← OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- void downsample (vec08u *apDest, const vec08u *apcSrc, int aOutBlockWidth, int aOutBlockHeight, int a
 —
 InBlockStride, int aOutBlockStride)

x2 downsampling.

 void downsample_16u (vec16u *apDest, const vec16u *apcSrc, int alnBlockWidth, int alnBlockHeight, int aOutBlockWidth, int aOutBlockHeight, int alnBlockStride, int aOutBlockStride)

x2 downsampling, 16-bit.

void downsample_gauss (vec08u *apDest, const vec08u *apcSrc, int32s aOutBlockWidth, int32s aOut
 BlockHeight, int32s aInBlockStride, int32s aOutBlockStride)

x2 downsampling using Gaussian blur.

5.49.2 Function Documentation

```
5.49.2.1 KERNEL_INFO apu_downsample ( " apu_downsample " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image 2x downsampling kernel metadata.

Parameters

DOWNSAMPL↔	Define for Kernel name
E_KN	
2	Number of ports
Port DOWNSA←	Define for name of input image (unsigned 8bit)
MPLE_KN_IN	
Port	Define for name of output image (unsigned 8bit)
DOWNSAMPL⇔	
E_KN_OUT	

```
5.49.2.2 KERNEL_INFO apu_downsample_16u ( "apu_downsample_16u ", 2, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image 2x downsampling kernel (16-bit version) metadata.

Parameters

DOWNSAMPL↔	Define for Kernel name
E_16u_KN	
2	Number of ports
Port DOWNSA←	Define for name of input image (unsigned 16bit)
MPLE_KN_IN	
Port	Define for name of output image (unsigned 16bit)
DOWNSAMPL⇔	
E_KN_OUT	

```
5.49.2.3 KERNEL_INFO apu_downsample_gauss ( " apu_downsample_gauss " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Image 2x downsampling using Gaussian blur kernel metadata.

Parameters

DOWNSAMPL↔	Define for Kernel name
E_GAUSS_KN	
2	Number of ports
Port DOWNSA←	Define for name of input image (unsigned 8bit)
MPLE_KN_IN	
Port	Define for name of output image (unsigned 8bit)
DOWNSAMPL⇔	
E_KN_OUT	

5.49.2.4 void downsample (vec08u * apDest, const vec08u * apcSrc, int aOutBlockWidth, int aOutBlockHeight, int aInBlockStride, int aOutBlockStride)

x2 downsampling.

Downsamples the image by two.

Parameters

apDest	- [Output] pointer to the destination buffer
apcSrc	- [Input] pointer to the source buffer
aOutBlockWidth	- [Input] width of one output data tile
aOutBlockHeight	- [Input] height of one output data tile
alnBlockStride	- [Input] line stride of the source data
aOutBlockStride	- [Input] line stride of the destination data

5.49.2.5 void downsample_16u (vec16u * apDest, const vec16u * apcSrc, int alnBlockWidth, int alnBlockHeight, int aOutBlockWidth, int aOutBlockHeight, int alnBlockStride, int aOutBlockStride)

x2 downsampling, 16-bit.

Downsamples the image by two. 16-bit version

Parameters

apDest	- [Output] pointer to the destination buffer
apcSrc	- [Input] pointer to the source buffer
aOutBlockWidth	- [Input] width of one output data tile
aOutBlockHeight	- [Input] height of one output data tile
alnBlockStride	- [Input] line stride of the source data
aOutBlockStride	- [Input] line stride of the destination data

5.49.2.6 void downsample_gauss (vec08u * apDest, const vec08u * apcSrc, int32s aOutBlockWidth, int32s aOutBlockStride, int32s aOutBlockStride)

x2 downsampling using Gaussian blur.

Downsamples the image by two using Gaussian blur.

apDest	- [Output] pointer to the destination buffer
apcSrc	- [Input] pointer to the source buffer
aOutBlockWidth	- [Input] width of one output data tile
aOutBlockHeight	- [Input] height of one output data tile
alnBlockStride	- [Input] line stride of the source data
aOutBlockStride	- [Input] line stride of the destination data

5.50 Image upsampling

5.50.1 Detailed Description

Image upsampling.

Collaboration diagram for Image upsampling:



Functions

- KERNEL_INFO apu_upsample (" apu_upsample ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)))
- void upsample (vec08u *apDest, const vec08u *apcSrc, int alnBlockWidth, int alnBlockHeight, int alnBlockHeight, int alnBlockStride)

x2 upsampling.

5.50.2 Function Documentation

```
5.50.2.1 KERNEL_INFO apu_upsample ( " apu_upsample " , 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)) )
```

Image 2x upsampling kernel metadata.

Parameters

UPSAMPLE_KN	Define for Kernel name
2	Number of ports
Port UPSAMP↔	Define for name of input image (unsigned 8bit)
LE_KN_IN	
Port UPSAMP↔	Define for name of output image (unsigned 8bit)
LE_KN_OUT	

5.50.2.2 void upsample (vec08u * apDest, const vec08u * apcSrc, int alnBlockWidth, int alnBlockHeight, int alnBlockStride, int aOutBlockStride)

x2 upsampling.

Upsamples the image by two.

apDest	- [Output] pointer to the destination buffer
apcSrc	- [Input] pointer to the source buffer
alnBlockWidth	- [Input] width of one input data tile
alnBlockHeight	- [Input] height of one input data tile
alnBlockStride	- [Input] line stride of the source data
aOutBlockStride	- [Input] line stride of the destination data

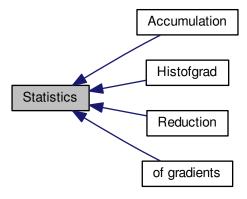
5.51 Statistics 189

5.51 Statistics

5.51.1 Detailed Description

sum of columns thru each row

Collaboration diagram for Statistics:



Modules

Accumulation

Element value accumulation.

Histofgrad

Histogram of Gradient.

· of gradients

Histogram.

Reduction

Vector to scalar reduction.

Functions

KERNEL_INFO apu_columns_sum (" apu_columns_sum ", 6, __port(__index(0), __identifier("INPUT_\circ 0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_\circ size(1, 1), __ek_size(4, 10)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VE\circ C_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(2), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), \circ __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), \circ __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_2"), __\circ attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_\circ size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("INPUT_3"), __attributes(ACF_ATTR_SCL_IN\circ STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

5.51.2 Function Documentation

```
5.51.2.1 KERNEL_INFO apu_columns_sum ( "apu_columns_sum ", 6 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 10)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(2), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(3), __identifier("INPUT_1"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(5), __identifier("INPUT_3"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Histogram kernel metadata.

5.51 Statistics

HISTOGRAM_←	Define for Kernel name
KN	
2	Number of ports
Port HISTOGR←	Define for name of input image (unsigned 8bit)
AM_KN_IN	
Port HISTOGR←	Define for name of histogram vector output, CU count X 256 (unsigned 32bit)
AM_KN_OUT	

5.52 Accumulation

5.52.1 Detailed Description

Element value accumulation.

Collaboration diagram for Accumulation:



Macros

- #define ACCUM_TILE_SIZE_X 10
- #define ACCUM TILE SIZE Y 10

Functions

- KERNEL_INFO apu_accumulation (" apu_accumulation ", 6, __port(_index(0), __identifier("INPUT_Img"), _attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ ⇔ ek_size(10, 10)), __port(_index(1), __identifier("Output_Img"), __attributes(ACF_ATTR_VEC_OUT), __ ⇔ spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __ ⇔ identifier("ACCUM_XOFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("ACCUM_YOF ⇔ FS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("ACCUM_XWIDTH"), __attributes(ACF_ ⇔ ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek ⇔ __size(1, 1)), __port(_index(5), __identifier("ACCUM_YHEIGHT"), __attributes(ACF_ATTR_SCL_IN_STA ⇔ TIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1))
- void accumulation_in32s_filter (vec32s *dst, vec32s *srcA, int16s sstr, int16s xOffs, int16s yOffs, int16s x←
 AccWidth, int16s yAccHeight)

Accumulates all values in a chunk (signed 32bit).

void accumulation_in32u_filter (vec32u *lpvOut, vec32u *lpvIn, int16s strideWidth, int16s xOffs, int16s yOffs, int16s xAccWidth, int16s yAccHeight)

Accumulates all values in a chunk (unsigned 32bit).

5.52.2 Function Documentation

5.52.2.1 void accumulation_in32s_filter (vec32s * dst, vec32s * srcA, int16s sstr, int16s xOffs, int16s yAccWidth, int16s yAccHeight)

Accumulates all values in a chunk (signed 32bit).

Accumulates the sum of all signed 32bit values in a chunk. Input is a vector/matrix, output is one sum-value

5.52 Accumulation 193

Parameters

dst	- [Output] pointer to output accumulation value
srcA	- [Input] Source block pointer of img A
sstr	- [Input] Source block width in elements (including padding)
xOffs	- [Input] X Offset where to start accumulation
yOffs	- [Input] Y Offset where to start accumulation
xAccWidth	- [Input] Width inside block for which accumulation has to be performed
yAccHeight	- [Input] Height inside block for which accumulation has to be performed

5.52.2.2 void accumulation_in32u_filter (vec32u * *lpvOut*, vec32u * *lpvIn*, int16s *strideWidth*, int16s *xOffs*, int16s *yOffs*, int16s *xAccWidth*, int16s *yAccHeight*)

Accumulates all values in a chunk (unsigned 32bit).

Accumulates the sum of all unsigned 32bit values in a chunk. Input is a vector/matrix, output is one sum-value

Parameters

dst	- [Output] pointer to output accumulation value
srcA	- [Input] Source block pointer of img A
sstr	- [Input] Source block width in elements (including padding)
xOffs	- [Input] X Offset where to start accumulation
yOffs	- [Input] Y Offset where to start accumulation
xAccWidth	- [Input] Width inside block for which accumulation has to be performed
yAccHeight	- [Input] Height inside block for which accumulation has to be performed

```
5.52.2.3 KERNEL_INFO apu_accumulation ( "apu_accumulation ", 6, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(10, 10)), __port(_index(1), __identifier("Output_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("ACCUM_XOFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("ACCUM_YOFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("ACCUM_XWIDTH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("ACCUM_YHEIGHT"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1))
```

Accumulation kernel metadata. Accumulates the pixels in one a certain neighborhood (XWIDTH, YHEIGHT) of signed 32bit image according to the chunk related xoffs and yoffs Outputs 16bit signed interpolation vector having as many elements as number of chunks .

the accumulation is not crossing the chunk limits.

Parameters

ACCUM_KN	Define for Kernel name
6	Number of ports
Port ACCUM_IN	Define for name offirst input image (signed 32bit)
Port	Define for name of interpolation result of the input image (signed 32bit).
ACCUM_OUT	
Port ACCUML←	Define for name of signed 16bit x offset vector (has one element for each chunk of the image)
_XOFFS	(signed 16bit)

194 Module Documentation

Port ACCUM_←	Define for name of signed 16bit y offset vector (has one element for each chunk of the image)
YOFFS	(signed 16bit)
Port ACCUM_←	Define for name of scalar value defining the accumulation width (has to be defined by the
XWIDTH	user) (signed 16bit)
Port ACCUM_←	Define for name of scalar value defining the accumulation height (has to be defined by the
YHEIGHT	user) (signed 16bit)

5.53 Column_sum 195

5.53 Column_sum

5.53.1 Detailed Description

Functions

void column_sum (vec08u *lpvIn0, vec32u *lpvOutDown, vec32u *lpvOutUp, bool isFirstTile, int IStrideIn0, int chunkWidth, int chunkHeight, int outChunkWidth, int priorityDown, int priorityUp, int indexOfTileStart)
 Elementwise unsigned 8bit addition => unsigned 16bit.

5.53.2 Function Documentation

5.53.2.1 void column_sum (vec08u * IpvIn0, vec32u * IpvOutDown, vec32u * IpvOutUp, bool isFirstTile, int IStrideIn0, int chunkWidth, int chunkHeight, int outChunkWidth, int priorityDown, int priorityUp, int indexOfTileStart)

Elementwise unsigned 8bit addition => unsigned 16bit.

Parameters

lpvln0	- [Input] 8bit source block pointer of img 0
<i>lpvOutDown</i>	- [Output] 32 unsigned bit destination block pointer
lpvOutUp	- [Output] 32 unsigned bit destination block pointer
isFirstTile	- [Input] boolean: true if the algorithm is at its first tile
lStride0	- [Input] Source block width (in bytes) including padding
chunkWidth	- [Input] Block width
chunkHeight	- [Input] Block height
outChunkWidth	- [Input] Block width
priorityDown	- [Input] Priority for lower part of input image
priorityUp	- [Input] Priority for upper part of input image
indexOfTileStart	- [Input] Index of tile start

196 Module Documentation

5.54 Histofgrad

5.54.1 Detailed Description

Histogram of Gradient.

Collaboration diagram for Histofgrad:



Functions

5.54.2 Function Documentation

```
5.54.2.1 KERNEL_INFO apu_histofgrad ( " apu_histofgrad " , 4 , __port(__index(0), __identifier("HOG_InGradX"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 4, 0, 4), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(4, 4)) , __port(__index(1), __identifier("HOG_InGradY"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 4, 0, 4), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(4, 4)) , __port(__index(2), __identifier("HOG_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size((16 *4 *1), 1), __ek_size(1, 1)) , __port(__index(3), __identifier("HOG_OUT_BINorm"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) )
```

Histogram of Gradient kernel metadata.

Parameters

HISTOFGRAD↔	Define for Kernel name
_KN	
3	Number of ports
Port HOG_KN←	Define for name of input gradientX image (signed 32bit)
_INGradX	
Port HOG_KN←	Define for name of input gradientY image (signed 32bit)
_INGradX	
Port	Define for name of histofgrad vector output, CU count X 64 bins of 4bits each (i.e. 2 x
HOG_KN_OUT	unsigned 32bit)

5.55 of gradients

5.55 of gradients

5.55.1 Detailed Description

Histogram.

Collaboration diagram for of gradients:



Macros

- #define HOG_NR_FEATURES_PER_HISTO 16
- #define HOG NR SCALES 1
- #define HOG_NR_OVERLAPPING_WINDOWS 4
- #define HOG_NR_FEATURES_PER_BOX (HOG_NR_FEATURES_PER_HISTO * HOG_NR_OVERLAP← PING_WINDOWS * HOG_NR_SCALES)
- #define HOG_OVERLAP 2
- #define HOG LAT DEPENDENCY 4
- #define HOG WND SZ 4

Functions

• void hog (vec08s *lpvInGradX, vec08s *lpvInGradY, vec16u *lpvOut, vec32u *lpvOutBINorm, int IStrideIn, int chunkWidth, int chunkHeight, int IStrideOut)

Elementwise unsigned 8bit addition => unsigned 16bit.

- KERNEL_INFO apu_histogram (" apu_histogram ", 2, __port(__index(0), __identifier("INPUT_0"), __ attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_ size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_STATIC_F IXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)))
- void hist (vec08u *lpvIn0, vec32u *lpvOut0, bool isFirstTile, int IStrideIn0, int chunkWidth, int chunkHeight, int outChunkWidth)

Elementwise unsigned 8bit addition => unsigned 16bit.

5.55.2 Function Documentation

```
5.55.2.1 KERNEL_INFO apu_histogram ( " apu_histogram " , 2 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)) )
```

Histogram kernel metadata.

198 Module Documentation

Parameters

HISTOGRAM_←	Define for Kernel name
KN	
2	Number of ports
Port HISTOGR←	Define for name of input image (unsigned 8bit)
AM_KN_IN	
Port HISTOGR←	Define for name of histogram vector output, CU count X 256 (unsigned 32bit)
AM_KN_OUT	

5.55.2.2 void hist (vec08u * *IpvIn0*, vec32u * *IpvOut0*, bool *isFirstTile*, int *IStrideIn0*, int *chunkWidth*, int *chunkWidth*, int *chunkWidth*)

Elementwise unsigned 8bit addition => unsigned 16bit.

Histogram computation of an input image

Parameters

lpvln0	- [Input] 8bit source block pointer of img 0
lpvOut0	- [Output] 32 unsigned bit destination block pointer
isFirstTile	- [Input] boolean: true if the algorithm is at its first tile
lStride0	- [Input] Source block width (in bytes) including padding
chunkWidth	- [Input] Block width
chunkHeight	- [Input] Block height

5.55.2.3 void hog (vec08s * IpvInGradX, vec08s * IpvInGradY, vec16u * IpvOut, vec32u * IpvOutBINorm, int IStrideIn, int chunkWidth, int chunkHeight, int IStrideOut)

Elementwise unsigned 8bit addition => unsigned 16bit.

Histogram computation of an input image

Parameters

lpvInGradX	- [Input] signed integral image of gradient X
lpvInGradY	- [Input] signed integral image of gradient Y
lpvOut	- [Output] histogram of gradients of 4x4 blocks on 2 different scales stored on 4 32bit words
isFirstTile	- [Input] boolean: true if the algorithm is at its first tile
<i>IStrideIn</i>	- [Input] Source block width (in elements) including padding
chunkWidth	- [Input] Block width
chunkHeight	- [Input] Block height
<i>IStrideOut</i>	- [Input] Destination block width (in elements) including padding

5.56 Reduction 199

5.56 Reduction

5.56.1 Detailed Description

Vector to scalar reduction.

Collaboration diagram for Reduction:



Functions

- KERNEL_INFO apu_reduction (" apu_reduction ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_
 size(1, 1), __ek_size(256, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SC
 L_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)))
- void reduc (vec32u *lpvIn0, int32s *lpOut0, bool isLastTile, int16s lFirstCuld, int16s lTileWidthInChunks, int lChunkWidth, int lChunkHeight, int lChunkSpanIn0, int lChunkSpanOut0)

Elementwise unsigned 8bit addition => unsigned 16bit.

- KERNEL_INFO apu_reduction_for_clmn (" apu_reduction_for_clmn ", 4, __port(__index(0), __identifier("I\cop\ NPUT_0"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_\cop\ type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(AC\cop\ F_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __\cop\ ek_size(4, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SCL_OUT_STATIC\cop\ FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)), __port(__\cop\ index(3), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)))
- void reduc (vec32u *lpvIn0, vec32u *lpvIn1, int32s *lpOut0, int32s *lpOut1, bool isLastTile, int16s lFirstCuld, int16s lTileWidthInChunks, int lChunkWidth, int lChunkHeight, int lChunkSpanIn0, int lChunkSpanOut0)

Elementwise unsigned 8bit addition => unsigned 16bit.

5.56.2 Function Documentation

```
5.56.2.1 KERNEL_INFO apu_reduction ( "apu_reduction ", 2, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)) )
```

256 vectors to 256 scalars accumulation reduction.

Parameters

ſ	DEDUCTION	Define for Keynel ware
	$REDUCTION_{\leftarrow}$	Define for Kernel name
	KN	
- 1	7.0.4	

200 Module Documentation

2	Number of ports
Port REDUCTI←	Define for name of vector input (unsigned 32bit)
ON_KN_IN	
Port REDUCTI←	Define for name of scalar output (unsigned 32bit)
ON KN OUT	

```
5.56.2.2 KERNEL_INFO apu_reduction_for_clmn ( "apu_reduction_for_clmn ", 4 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)) , __port(_index(3), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)) )
```

Parameters

REDUCTION_←	Define for Kernel name
KN	
2	Number of ports
Port REDUCTI←	Define for name of vector input (unsigned 32bit)
ON_KN_IN	
Port REDUCTI←	Define for name of scalar output (unsigned 32bit)
ON_KN_OUT	

5.56.2.3 void reduc (vec32u * *IpvIn0*, vec32u * *IpvIn1*, int32s * *IpOut0*, int32s * *IpOut1*, bool *isLastTile*, int16s *IFirstCuld*, int16s *ITileWidthInChunks*, int *IChunkWidth*, int *IChunkHeight*, int *IChunkSpanIn0*, int *IChunkSpanOut0*)

Elementwise unsigned 8bit addition => unsigned 16bit.

Reduce an input vector/image by summing up the corresponding elements

Parameters

lpvIn0	- [Input] 32unsigned bit source block pointer of img 0
lpOut0	- [Output] 32signed bit destination block pointer
isLastTile	- [Input] boolean: is true, if the last image tile is being processed
<i>IFirstCuld</i>	- [Input] the id of the first CU
lTileWidthIn⇔	- [Input] number of used chunks/CUs
Chunks	
chunkWidth	- [Input] Block width
chunkHeight	- [Input] Block height
IChunkSpanIn0-	[Input] Source block width (in bytes) including padding
lChunkSpan⇔	[Input] Destination block width (in bytes) including padding
Out0-	

5.56.2.4 void reduc (vec32u * *IpvIn0*, int32s * *IpOut0*, bool *isLastTile*, int16s *IFirstCuld*, int16s *ITileWidthInChunks*, int *IChunkWidth*, int *IChunkHeight*, int *IChunkSpanIn0*, int *IChunkSpanOut0*)

Elementwise unsigned 8bit addition => unsigned 16bit.

Reduce an input vector/image by summing up the corresponding elements

5.56 Reduction 201

Parameters

lpvln0	- [Input] 32unsigned bit source block pointer of img 0
lpOut0	- [Output] 32signed bit destination block pointer
isLastTile	- [Input] boolean: is true, if the last image tile is being processed
<i>IFirstCuld</i>	- [Input] the id of the first CU
lTileWidthIn⇔	- [Input] number of used chunks/CUs
Chunks	
chunkWidth	- [Input] Block width
chunkHeight	- [Input] Block height
IChunkSpanIn0-	[Input] Source block width (in bytes) including padding
IChunkSpan⇔	[Input] Destination block width (in bytes) including padding
Out0-	



Chapter 6

Class Documentation

6.1 APEX_HaarCascadeFeature Struct Reference

Public Attributes

- int16u threshold
- int16u leftVal
- int16u rightVal
- int16u x5y5w5_1
- int08u h5type3

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

6.2 APEX_HaarCascadeStage Struct Reference

Public Attributes

- int08u featureCount
- int16u thresholdFixed

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

6.3 APEX_lbpFeature Struct Reference

Public Attributes

- int32_t values [8]
- STAGE_FIXED_POINT_TYPE_SCALAR leafValuesFixed [2]
- uint8 t x
- uint8_t **y**
- uint8_t width

204 Class Documentation

• uint8_t height

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

6.4 APEX_lbpStage Struct Reference

Public Attributes

- uint8 t featureCount
- STAGE_FIXED_POINT_TYPE_SCALAR threshold

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

6.5 point_t Struct Reference

Public Attributes

- int x
- int y

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

6.6 rect_t Struct Reference

Public Attributes

- int16u left
- int16u **top**
- int16u width
- int16u height

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

6.7 RESIZE DESCRIPTOR Struct Reference

Struct for Resize descriptor.

#include <resize_definitions_apu.h>

Public Attributes

- · int32_t dst_bh
- int32_t dst_bw
- int32_t src_current_line
- · int32 t unused
- int32_t src_offs
- int32_t scl_fact
- int32_t phases
- · int32_t taps
- int32_t out_scale
- · int32_t out_round
- const int16 t * fbnk

6.7.1 Detailed Description

Struct for Resize descriptor.

Sizes and Offsets are on vertical direction (Y)

NB: src_current_line, src_offs cleared by a call to get_polyphase_params();

6.7.2 Member Data Documentation

6.7.2.1 int32_t RESIZE_DESCRIPTOR::dst_bw

Destination block width and height

6.7.2.2 const int16_t * RESIZE_DESCRIPTOR::fbnk

Polyphase Filter Bank

6.7.2.3 int32_t RESIZE_DESCRIPTOR::out_round

Values for scaling and rounding

6.7.2.4 int32_t RESIZE_DESCRIPTOR::scl_fact

Source offset - Scale Factor

6.7.2.5 int32_t RESIZE_DESCRIPTOR::taps

Number of taps and phases

6.7.2.6 int32 t RESIZE DESCRIPTOR::unused

Line number of the top left corner of the source (input) block.

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

206 Class Documentation

6.8 resize_descriptor Struct Reference

Public Attributes

- int32_t src_offs
- int32_t scl_fact
- int16_t phases
- int16_t taps
- int32_t out_scale
- int32_t out_round
- int16_t * fbnk

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

Chapter 7

File Documentation

7.1 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_gdc_ldw2/src/cg_kernel.h File Reference

Contains the prototypes for the APU kernels found in cg_kernel.a.

7.1.1 Detailed Description

Contains the prototypes for the APU kernels found in cg_kernel.a.

7.2 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_remap/src/cg_kernel.h File Reference

Contains the prototypes for the APU kernels found in cg_kernel.a.

7.2.1 Detailed Description

Contains the prototypes for the APU kernels found in cg_kernel.a.

7.3 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_resize/src/cg_kernel.h File Reference

Contains the prototypes for the APU kernels found in cg_kernel.a.

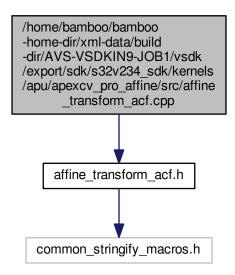
7.3.1 Detailed Description

Contains the prototypes for the APU kernels found in cg_kernel.a.

7.4 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_acf.cpp File Reference

ACF metadata and wrapper function for the affine transformation.

#include "affine_transform_acf.h"
Include dependency graph for affine_transform_acf.cpp:



Functions

KERNEL_INFO affine_bilinear_interpolate (" affine_bilinear_interpolate ", 5, __port(__index(0), __ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MATRIX"), __attributes(ACF_ATTR_SC_CL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(6, 1), __ek_size(1, 1)), __port(__index(2), __identifier("IMAGE_WIDTH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("IMAGE_HEIGHT"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

ACF metadata for the bilinear interpolation.

7.4.1 Detailed Description

ACF metadata and wrapper function for the affine transformation.

7.4.2 Function Documentation

7.5 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_acf.h File
Reference

7.4.2.1 KERNEL_INFO affine_bilinear_interpolate ("affine_bilinear_interpolate ", 5 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("MATRIX"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(6, 1), __ek_size(1, 1)) , __port(_index(2), __identifier("IMAGE_WIDTH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(3), __identifier("IMAGE_HEIGHT"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(4), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

ACF metadata for the bilinear interpolation.

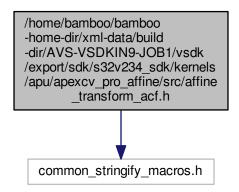
See also

UG-10267-03 ACF User Guide, Section 3.2.2

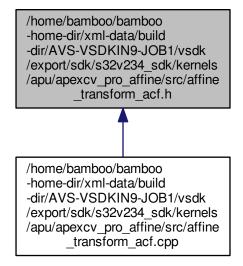
7.5 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_acf.h File Reference

ACF metadata and wrapper function for the affine transformation.

#include "common_stringify_macros.h"
Include dependency graph for affine_transform_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define AFFINE_BILINEAR_INTERPOLATE_K affine_bilinear_interpolate
- #define AFFINE_BILINEAR_INTERPOLATE_KN XSTR(AFFINE_BILINEAR_INTERPOLATE_K)
- #define INPUT "INPUT"
- #define MATRIX "MATRIX"
- #define IMAGE WIDTH "IMAGE WIDTH"
- #define IMAGE HEIGHT "IMAGE HEIGHT"
- #define **OUTPUT** "OUTPUT"

Functions

• extKernelInfoDecl (affine_bilinear_interpolate)

7.5.1 Detailed Description

ACF metadata and wrapper function for the affine transformation.

7.6 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_affine/src/affine_transform_apu.cpp File Reference

ACF Affine Transform Wrapper.

7.6.1 Detailed Description

ACF Affine Transform Wrapper.

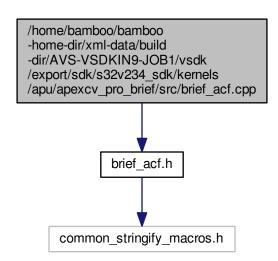
Reference 211

APU affine transform kernel.

7.7 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_brief/src/brief_acf.cpp File Reference

ACF metadata and wrapper function for the BRIEF.

#include "brief_acf.h"
Include dependency graph for brief_acf.cpp:



Functions

• KERNEL_INFO compute_brief_descriptor (" compute_brief_descriptor ", 11, __port(__index(0), __ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), e0 size(1, 1), ek size(36, 36)), port(index(1), identifier("FILTER TYPE"), attributes(AC← F_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), _ek_size(1, 1)), __port(__index(2), __identifier("SMPL_PACKET"), __attributes(ACF_ATTR_SCL_IN_ --STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(2048, 1)), __port(__index(3), __identifier("NR_PACKETS_UL"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("NR_PACKETS_UR"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, $0, \ 0), \ \underline{\hspace{0.1cm}} = 0 \underline{\hspace{0.1cm}} ata\underline{\hspace{0.1cm}} type(d32u), \ \underline{\hspace{0.1cm}} = 0 \underline{\hspace{0.1cm}} size(1, \ 1), \ \underline{\hspace{0.1cm}} ek\underline{\hspace{0.1cm}} size(1, \ 1)), \ \underline{\hspace{0.1cm}} port(\underline{\hspace{0.1cm}} index(5), \ \underline{\hspace{0.1cm}} identifier("NR_\longleftrightarrow R.))$ PACKETS_LL"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("NR_PACKETS_LR"), _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __ <-e0 size(1, 1), ek size(1, 1)), port(index(7), identifier("PATCH SIZE"), attributes(ACF ATTR ← SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, $1)), \ \underline{\quad} port(\underline{\quad} index(8), \ \underline{\quad} identifier("DESC_SIZE"), \ \underline{\quad} attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), \ \leftarrow \\ \underline{\quad} identifier("DESC_SIZE"), \ \underline{\quad} attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), \ \leftarrow \\ \underline{\quad} identifier(\underline{\quad} index(8), \ \underline{\quad} index(8), \ \underline{\quad$ __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(9), identifier("COUNT"), attributes(ACF ATTR SCL OUT STATIC FIXED), spatial dep(0, 0, 0, 0), OR_OUT"), __attributes(ACF_ATTR_VEC_OUT_FIFO_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ ~ type(d08u), __e0_size(1, 1), __ek_size(64, 1)))

ACF metadata for the computation of BRIEF descriptors.

7.7.1 Detailed Description

ACF metadata and wrapper function for the BRIEF.

7.7.2 Function Documentation

```
7.7.2.1 KERNEL INFO compute brief descriptor ( "compute brief descriptor ", 11, __port(_index(0), _identifier("INPUT"),
        __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(36, 36)) ,
          _port(__index(1), __identifier("FILTER_TYPE"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0,
        0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("SMPL_PACKET"),
        __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1),
         _ek_size(2048, 1)), _port(_index(3), _identifier("NR_PACKETS_UL"), _attributes(ACF_ATTR_SCL_IN_STATI
        C_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(4),
          identifier("NR PACKETS UR"), attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, 0, 0, 0),
          _e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(5), __identifier("NR_PACKETS_LL"),
        __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1),
        _ek_size(1, 1)), _port(_index(6), _identifier("NR_PACKETS_LR"), _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED),
        __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(7),
        __identifier("PATCH_SIZE"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0),
        _e0_data_type(d08u), _e0_size(1, 1), _ek_size(1, 1)) , _port(_index(8), _identifier("DESC_SIZE"),
        _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), _e0_size(1, 1),
        __ek_size(1, 1)) , __port(_index(9), __identifier("COUNT"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED),
        __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(10),
        __identifier("DESCRIPTOR_OUT"), __attributes(ACF_ATTR_VEC_OUT_FIFO_FIXED), __spatial_dep(0, 0, 0, 0),
        __e0_data_type(d08u), __e0_size(1, 1), __ek_size(64, 1)) )
```

ACF metadata for the computation of BRIEF descriptors.

See also

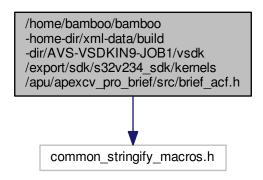
UG-10267-03 ACF User Guide, Section 3.2.2

7.8 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_brief/src/brief_acf.h File Reference

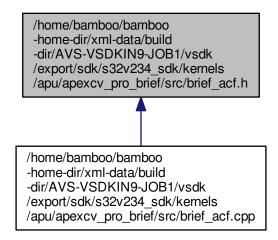
ACF metadata for the BRIEF.

Reference 213

#include "common_stringify_macros.h"
Include dependency graph for brief_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define COMPUTE_BRIEF_DESCRIPTOR_K compute_brief_descriptor
- #define COMPUTE_BRIEF_DESCRIPTOR_KN XSTR(COMPUTE_BRIEF_DESCRIPTOR_K)
- #define INPUT "INPUT"
- #define FILTER_TYPE "FILTER_TYPE"
- #define SMPL_PACKET "SMPL_PACKET"
- #define NR PACKETS UL "NR PACKETS UL"
- #define NR PACKETS UR "NR PACKETS UR"
- #define NR_PACKETS_LL "NR_PACKETS_LL"

- #define NR_PACKETS_LR "NR_PACKETS_LR"
- #define PATCH_SIZE "PATCH_SIZE"
- #define DESC_SIZE "DESC_SIZE"
- #define COUNT "COUNT"
- #define DESCRIPTOR_OUT "DESCRIPTOR_OUT"

Functions

• extKernelInfoDecl (compute brief descriptor)

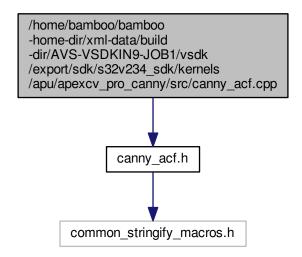
7.8.1 Detailed Description

ACF metadata for the BRIEF.

7.9 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234-_sdk/kernels/apu/apexcv_pro_canny/src/canny_acf.cpp File Reference

ACF Metadata and wrapper function for Canny edge detector.

```
#include "canny_acf.h"
Include dependency graph for canny acf.cpp:
```



Functions

KERNEL_INFO canny_non_maxima_suppress (" canny_non_maxima_suppress ", 4, __port(__index(0), ← __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(1), __identifier("LOW_THRESH"), __attributes(ACF_A← TTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek← __size(1, 1)), __port(__index(2), __identifier("HIGH_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC← __FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__←

data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)))

ACF metadata for the non-maxima suppression kernel.

KERNEL_INFO canny_nms_promote (" canny_nms_promote ", 4, __port(__index(0), __identifier("INPU←T"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)), __port(_index(1), __identifier("LOW_THRESH"), __attributes(ACF_ATTR_SCL_IN_ST←ATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(← __index(2), __identifier("HIGH_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_← dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("O←UTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(2, 1)))

ACF metadata for the non-maxima suppression & edge promotion kernel.

KERNEL_INFO canny_promote_edges (" canny_promote_edges ", 2, __port(__index(0), __identifier("IN←PUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __←spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)))

ACF metadata for the edge promotion kernel.

KERNEL_INFO canny_promote_edges_full (" canny_promote_edges_full ", 2, __port(_index(0), __
identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u),
 _e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_V
EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)))

ACF metadata for the internal edge promotion kernel.

KERNEL_INFO canny_create_image (" canny_create_image ", 2, __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
ek_size(2, 2)), __port(_index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial
__dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)))

ACF metadata for the create image kernel.

7.9.1 Detailed Description

ACF Metadata and wrapper function for Canny edge detector.

7.9.2 Function Documentation

7.9.2.1 KERNEL_INFO canny_create_image (" canny_create_image ", 2, __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)), __port(_index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)))

ACF metadata for the create image kernel.

7.9.2.2 KERNEL_INFO canny_nms_promote (" canny_nms_promote " , 4 , __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)) , __port(__index(1), __identifier("LOW_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("HIGH_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)))

ACF metadata for the non-maxima suppression & edge promotion kernel.

```
7.9.2.3 KERNEL_INFO canny_non_maxima_suppress ( " canny_non_maxima_suppress " , 4 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)) , __port(__index(1), __identifier("LOW_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("HIGH_THRESH"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 1)) )
```

ACF metadata for the non-maxima suppression kernel.

```
7.9.2.4 KERNEL_INFO canny_promote_edges ( " canny_promote_edges " , 2 , __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)) )
```

ACF metadata for the edge promotion kernel.

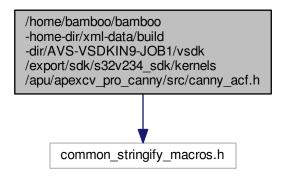
```
7.9.2.5 KERNEL_INFO canny_promote_edges_full ( " canny_promote_edges_full " , 2 , __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)) , __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)) )
```

ACF metadata for the internal edge promotion kernel.

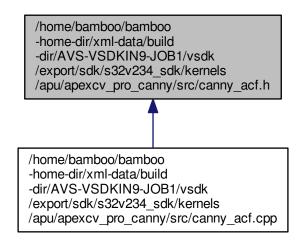
7.10 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_canny/src/canny_acf.h File Reference

ACF Metadata for Canny edge detector.

```
#include "common_stringify_macros.h"
Include dependency graph for canny_acf.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define CANNY NON MAXIMA SUPPRESS K canny non maxima suppress
- #define CANNY_NON_MAXIMA_SUPPRESS_KN XSTR(CANNY_NON_MAXIMA_SUPPRESS_K)
- #define CANNY NMS PROMOTE K canny nms promote
- #define CANNY NMS PROMOTE KN XSTR(CANNY NMS PROMOTE K)
- #define CANNY_PROMOTE_EDGES_K canny_promote_edges
- #define CANNY_PROMOTE_EDGES_KN XSTR(CANNY_PROMOTE_EDGES_K)
- #define CANNY_PROMOTE_EDGES_FULL_K canny_promote_edges_full
- #define CANNY PROMOTE EDGES FULL KN XSTR(CANNY PROMOTE EDGES FULL K)
- #define CANNY CREATE IMAGE K canny create image
- #define CANNY_CREATE_IMAGE_KN XSTR(CANNY_CREATE_IMAGE_K)
- #define INPUT "INPUT"
- #define LOW_THRESH "LOW THRESH"
- #define HIGH_THRESH "HIGH_THRESH"
- #define **OUTPUT** "OUTPUT"

Functions

- extKernelInfoDecl (canny_non_maxima_suppress)
- extKernelInfoDecl (canny nms promote)
- extKernelInfoDecl (canny promote edges)
- extKernelInfoDecl (CANNY_PROMOTE_EDGES_ITERATIONS_K)
- extKernelInfoDecl (canny_promote_edges_full)
- extKernelInfoDecl (canny create image)

7.10.1 **Detailed Description**

ACF Metadata for Canny edge detector.

7.11 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_canny/src/canny_apu.cpp File Reference

Canny Edge Detection Kernels.

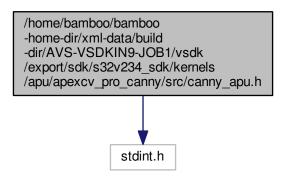
7.11.1 Detailed Description

Canny Edge Detection Kernels.

7.12 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/apexcv_pro_canny/src/canny_apu.h File Reference

Canny Edge Detection Kernels.

#include <stdint.h>
Include dependency graph for canny_apu.h:



Macros

- #define MAX_BLK_SIZE 96
- #define MAX_STACK_SIZE (MAX_BLK_SIZE/2)
- #define ROW_BIT_OFFSET 8

Functions

- void apu_canny_suppress (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh, int16u low, int16u high)
- void apu_canny_suppress_promote (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int16u low, int16u high)
- void apu_canny_connect_edges (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)
- void apu canny connect edges full (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)
- void apu_canny_create_image (vec08u *dst, int dstr, const vec08u *edge, int estr, int bw, int bh)

sdk/kernels/apu/apexcv_pro_canny/src/canny_apu.h File
Reference 219

7.12.1 Detailed Description

Canny Edge Detection Kernels.

7.12.2 Function Documentation

7.12.2.1 void apu_canny_connect_edges (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Performs edge connection between blocks. This is used where the block size does not change between kernel outputs

Parameters

dst	- [Output] Destination edge map buffer
dstr	- [Input] Destination edge map stride
src	- [Input] Source edge map buffer
sstr	- [Input] Source edge map stride
bw	- [Input] Block width
bh	- [Input] Block height

7.12.2.2 void apu_canny_connect_edges_full (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh)

Performs edge connection between blocks. This is used where the block size changes between kernel outputs

Parameters

dst	- [Output] Destination edge map buffer
dstr	- [Input] Destination edge map stride
src	- [Input] Source edge map buffer
sstr	- [Input] Source edge map stride
bw	- [Input] Block width
bh	- [Input] Block height

7.12.2.3 void apu_canny_create_image (vec08u * dst, int dstr, const vec08u * edge, int estr, int bw, int bh)

Outputs the src image with edge as a mask.

Parameters

dst	- [Output] Destination image buffer
dstr	- [Input] Destination image stride
src	- [Input] Source image buffer
sstr	- [Input] Source image stride
edge	- [Input] Source edge map buffer
estr	- [Input] Source edge map stride
bw	- [Input] Block width
bh	- [Input] Block height

7.12.2.4 void apu_canny_suppress (vec08u * dst, int dstr, const vec08u * src, int sstr, int bw, int bh, int16u low, int16u high

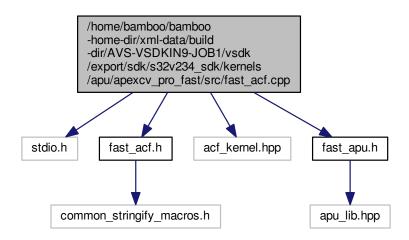
Performs Canny Non-Maxima Suppression and Edge Hysteresis

Parameters

dEdges	- [Output] Destination edge map buffer
estr	- [Input] Destination edge map stride
sMag	- [Input] Source magnitude buffer
mstr	- [Input] Source magnitude stride
dx	- [Input] Source X gradient buffer
xstr	- [Input] Source X gradient stride
dy	- [Input] Source Y gradient buffer
xstr	- [Input] Source Y gradient stride
bw	- [Input] Block width
bh	- [Input] Block height
low	- [Input] Low threshold used in edge hysteresis
high	- [Input] High threshold used in edge hysteresis

7.13 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_fast/src/fast_acf.cpp File Reference

```
#include <stdio.h>
#include "fast_acf.h"
#include "acf_kernel.hpp"
#include "fast_apu.h"
Include dependency graph for fast acf.cpp:
```



Functions

• KERNEL_INFO **fast_offset** (" fast_offset ", 3, __port(__index(0), __identifier("OUTPUT_OFFSETS"), __
attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __
e0_size(1, 1), __ek_size(16, 1)), __port(__index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_V
EC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__
index(2), __identifier("CIRCUMFERENCE"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial
__dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.14 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.cpp File Reference

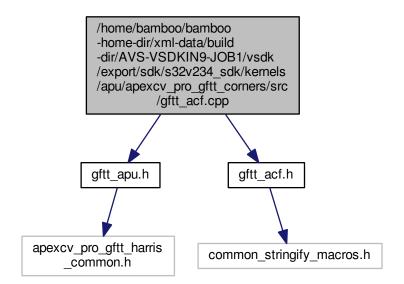
- KERNEL_INFO **fast** (" fast ", 4, __port(__index(0), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_\(\cup VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(\(\cup index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_\(\cup data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("THRESHOLD"), __\(\cup attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_\(\cup size(1, 1), __ek_size(1, 1), __ek_size(1, 1), __ek_size(1, 1), __ek_size(1, 1), __ek_size(16, 1)))
- KERNEL_INFO **fast_nms** (" fast_nms ", 4, __port(__index(0), __identifier("OUTPUT_0"), __attributes(AC \leftarrow F_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), \leftarrow __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("THRESHOLD"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __ \leftarrow e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OFFSET_TABLE"), __attributes(ACF_AT \leftarrow TR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_ \leftarrow size(16, 1)))
- KERNEL_INFO nms3x3 (" nms3x3 ", 2, __port(__index(0), __identifier("input"), __attributes(ACF_ATTR_\top VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_\top index(1), __identifier("output"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data\top type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.14 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.cpp File Reference

ACF metadata and wrapper function for the Good Features To Track.

```
#include "gftt_apu.h"
#include "gftt_acf.h"
```

Include dependency graph for gftt_acf.cpp:



Functions

- KERNEL INFO gftt_wrapper_box7_nms5 (" gftt_wrapper_box7_nms5 ", 12, __port(__index(0), __ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 1, 1), __e0_data_type(d08u), _e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("PARAMS"), __attributes(ACF_ATTR_← SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(6, $1)), \quad \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(2), \quad \underline{\hspace{0.5cm}} identifier("OUTPUT"), \quad \underline{\hspace{0.5cm}} attributes(ACF_ATTR_VEC_OUT), \quad \underline{\hspace{0.5cm}} spatial_dep(0, \underline{\hspace{0.5cm}} index(2), \underline{\hspace{0.5cm}} index(2$ 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 1)), __port(_index(3), __identifier("S← VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ \leftarrow type(d16s), _e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_7+1)), __port(__index(4), _identifier("SVXY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), ~~ $e0_data_type(d16s), \ \underline{\hspace{0.5cm}} e0_size(1,\ 1), \ \underline{\hspace{0.5cm}} ek_size(MAX_BLOCK_WIDTH,\ MAX_FILTER_Y_7+1)), \ \underline{\hspace{0.5cm}} port(\leftarrow 1,\ 1)$ _index(5), __identifier("SVYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), e0 data type(d16s), e0 size(1, 1), ek size(MAX BLOCK WIDTH, MAX FILTER Y ← _7+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), spatial dep(0, 0, 0, 0), e0 data type(d16s), e0 size(1, 1), ek size(MAX BLOCK WIDTH+MA↔ X_NMS_R-1, MAX_NMS_R+1)), __port(__index(7), __identifier("NMS"), __attributes(ACF_ATTR_VEC_O ← $\label{eq:ut_static_fixed} \mbox{UT_STATIC_FIXED}, \ __\mbox{spatial_dep(0, 0, 0, 0)}, \ __\mbox{e0_data_type(d16s)}, \ __\mbox{e0_size(1, 1)}, \ __\mbox{ek_size(MAX_} \mbox{\hookleftarrow} \mbox{\leftarrow} \mbox{$ $BLOCK_WIDTH,\ MAX_NMS_R+1)),\ __port(__index(8),\ __identifier("BXX"),\ __attributes(ACF_ATTR_VEC \leftarrow ATTR_VEC))$ AX_BLOCK_WIDTH+MAX_FILTER_X_7-1, 1)), __port(__index(9), __identifier("BXY"), __attributes(A← CF ATTR VEC OUT STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_7-1, 1)), __port(__index(10), __identifier("BYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), $_$ e0_size(1, 1), $_$ ek $_$ size(MAX $_$ BLOCK $_$ WIDTH+MAX $_$ FILTER $_$ X $_$ 7-1, 1)), $__$ port($_$ index(11), $__$ \hookleftarrow identifier("MAX_EIGEN"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO gftt_wrapper_box5_nms5 (" gftt_wrapper_box5_nms5 ", 12, __port(_index(0), __
 identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 1, 1), __e0_data_type(d08u),
 _e0_size(1, 1), __ek_size(4, 1)), __port(_index(1), __identifier("PARAMS"), __attributes(ACF_ATTR_
 SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(6,

Reference 22

- __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("S← VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_← type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_5+1)), __port(__index(4), _identifier("SVXY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __ e0 data type(d16s), e0 size(1, 1), ek size(MAX BLOCK WIDTH, MAX FILTER Y 5+1)), port(← _index(5), __identifier("SVYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), $__e0_data_type(d16s)$, $__e0_size(1, 1)$, $__ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y \leftrightarrow 1000 MeV.$ _5+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MA↔ X_NMS_R-1, MAX_NMS_R+1)), __port(__index(7), __identifier("NMS"), __attributes(ACF_ATTR_VEC_O ← $\label{eq:ut_static_fixed} \mbox{UT_STATIC_FIXED}, \ __\mbox{spatial_dep(0, 0, 0, 0)}, \ __\mbox{e0_data_type(d16s)}, \ __\mbox{e0_size(1, 1)}, \ __\mbox{ek_size(MAX_} \mbox{\leftarrow} \mbox{eV} \mbox{e$ BLOCK_WIDTH, MAX_NMS_R+1)), __port(__index(8), __identifier("BXX"), __attributes(ACF_ATTR_VEC← _OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(M↔ AX_BLOCK_WIDTH+MAX_FILTER_X_5-1, 1)), __port(__index(9), __identifier("BXY"), __attributes(A --CF ATTR VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), ek size(MAX BLOCK WIDTH+MAX FILTER X 5-1, 1)), port(index(10), identifier("BYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), $_e0_size(1,\ 1),\ __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_5-1,\ 1)),\ __port(__index(11),\ __\hookleftarrow 1)$ identifier("MAX_EIGEN"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), e0 data type(d16s), e0 size(1, 1), ek size(1, 1)))
- KERNEL_INFO gftt_wrapper_box3_nms5 (" gftt_wrapper_box3_nms5 ", 12, __port(__index(0), _ identifier("INPUT"), attributes(ACF ATTR VEC IN), spatial dep(2, 2, 1, 1), e0 data type(d08u), _e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("PARAMS"), __attributes(ACF_ATTR_← SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(6, $1)), \quad \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(2), \quad \underline{\hspace{0.5cm}} identifier("OUTPUT"), \quad \underline{\hspace{0.5cm}} attributes(ACF_ATTR_VEC_OUT), \quad \underline{\hspace{0.5cm}} spatial_dep(0, \underline{\hspace{0.5cm}} index(0, \underline{\hspace{0.5cm}} index(0,$ VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_3+1)), __port(__index(4), __identifier("SVXY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __ e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_3+1)), __port(← __index(5), __identifier("SVYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), $_$ e0_data_type(d16s), $_$ e0_size(1, 1), $_$ ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y \leftarrow _3+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MA X_NMS_R-1, MAX_NMS_R+1)), __port(__index(7), __identifier("NMS"), __attributes(ACF_ATTR_VEC_O UT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_← BLOCK WIDTH, MAX NMS R+1)), port(index(8), identifier("BXX"), attributes(ACF ATTR VEC _OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(M -AX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)), __port(__index(9), __identifier("BXY"), __attributes(A← CF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)), __port(__index(10), __identifier("BYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), $_$ e0 $_$ size(1, 1), $__$ ek $_$ size(MAX $_$ BLOCK $_$ WIDTH+MAX $_$ FILTER $_$ X $_$ 3-1, 1)), $__$ port($__$ index(11), $__$ \hookleftarrow identifier("MAX EIGEN"), attributes(ACF ATTR VEC OUT STATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d16s), e0 size(1, 1), ek size(1, 1)))
- KERNEL_INFO gftt_extract (" gftt_extract ", 8, __port(__index(0), __identifier("COORD"), __attributes(A← CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(MAX_CORNERS, 1)), __port(__index(1), __identifier("STREN"), __attributes(ACF_ATTR_SCL← OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(MA← X_CORNERS, 1)), __port(__index(2), __identifier("COUNT"), __attributes(ACF_ATTR_SCL_OUT_STATI← C_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(3), __identifier("LOCAL_COORD"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial← dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_CORNER_PER_CHUNK, 1)), ← __port(__index(4), __identifier("LOCAL_STREN"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_CORNER_PER_CH← UNK, 1)), __port(__index(5), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(6), __identifier("MAX_EIGE←

N"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(7), __identifier("PARAMS"), __attributes(ACF_ATTR_ \leftarrow SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(6, 1)))

KERNEL_INFO gftt_sort_and_filter (" gftt_sort_and_filter ", 7, __port(__index(0), __identifier("FEATUR←E"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_CORNERS *2, 1)), __port(__index(1), __identifier("FEAT_COUNT"), _← attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), _← e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INDEX"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(M←AX_CORNERS, 1)), __port(__index(3), __identifier("COORD"), __attributes(ACF_ATTR_SCL_IN_STATI←CFIXED), __ex_spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(MAX_CORNE←RS, 1)), __port(__index(4), __identifier("STREN"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __ex_spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(MAX_CORNERS, 1)), __port(←CIN_COUNT"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("PARAM←S"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __spatial_dep(0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __spatial_dep(0, 0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __spatial_dep(0, 0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __spatial_dep(0, 0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __spatial_dep(0, 0, 0, 0, 0, 0, __e0_data_type(d32s), __e0_size(1, 1), __spatial_dep(0, 0, 0, 0, 0, 0, 0, 0, 0, __e

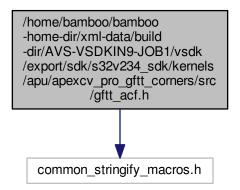
7.14.1 Detailed Description

ACF metadata and wrapper function for the Good Features To Track.

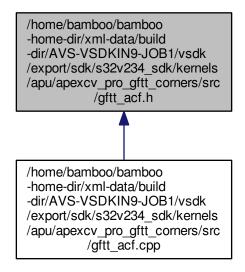
7.15 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_gftt_corners/src/gftt_acf.h File Reference

ACF metadata for the Good Features To Track.

#include "common_stringify_macros.h"
Include dependency graph for gftt_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define GFTT WRAPPER BOX7 NMS5 K gftt wrapper box7 nms5
- #define GFTT WRAPPER BOX7 NMS5 KN XSTR(GFTT WRAPPER BOX7 NMS5 K)
- #define GFTT_WRAPPER_BOX5_NMS5_K gftt_wrapper_box5_nms5
- #define GFTT WRAPPER BOX5 NMS5 KN XSTR(GFTT WRAPPER BOX5 NMS5 K)
- #define GFTT WRAPPER BOX3 NMS5 K gftt wrapper box3 nms5
- #define GFTT_WRAPPER_BOX3_NMS5_KN XSTR(GFTT_WRAPPER_BOX3_NMS5_K)
- #define GFTT EXTRACT K gftt extract
- #define GFTT_EXTRACT_KN XSTR(GFTT_EXTRACT_K)
- #define GFTT_SORT_AND_FILTER_K gftt_sort_and_filter
- #define GFTT_SORT_AND_FILTER_KN XSTR(GFTT_SORT_AND_FILTER_K)
- #define INPUT "INPUT"
- #define PARAMS "PARAMS"
- #define **OUTPUT** "OUTPUT"
- #define SVXX "SVXX"
- #define SVXY "SVXY"
- #define SVYY "SVYY"
- #define NMS_X "NMS_X"
- #define NMS "NMS"
- #define BXX "BXX"
- #define BXY "BXY"
- #define BYY "BYY"
- #define MAX_EIGEN "MAX_EIGEN"
- #define COORD "COORD"
- #define STREN "STREN"
- #define COUNT "COUNT"
- #define LOCAL COORD "LOCAL COORD"
- #define LOCAL STREN "LOCAL STREN"

- #define SRC "SRC"
- #define FEATURE "FEATURE"
- #define FEAT_COUNT "FEAT_COUNT"
- #define INDEX "INDEX"

Functions

- extKernelInfoDecl (gftt_wrapper_box7_nms5)
- extKernelInfoDecl (gftt wrapper box5 nms5)
- extKernelInfoDecl (gftt wrapper box3 nms5)
- extKernelInfoDecl (gftt extract)
- extKernelInfoDecl (gftt_sort_and_filter)

7.15.1 Detailed Description

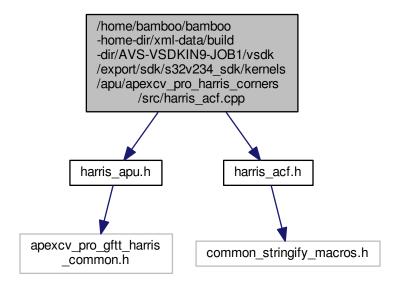
ACF metadata for the Good Features To Track.

7.16 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_harris_corners/src/harris_acf.cpp File Reference

ACF metadata and wrapper function for the harris corner.

```
#include "harris_apu.h"
#include "harris_acf.h"
```

Include dependency graph for harris_acf.cpp:



Functions

KERNEL_INFO harris_corners_box7_nms5 (" harris_corners_box7_nms5 ", 11, __port(__index(0), _
 __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 1, 1), __e0_data_type(d08u),

Reference 227

- $\underline{\hspace{0.5cm}}$ e0_size(1, 1), $\underline{\hspace{0.5cm}}$ ek_size(4, 1)), $\underline{\hspace{0.5cm}}$ port($\underline{\hspace{0.5cm}}$ index(1), $\underline{\hspace{0.5cm}}$ identifier("PARAMS"), $\underline{\hspace{0.5cm}}$ attributes(ACF_ATTR_ \longleftrightarrow SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(7, 1)), __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, $0, \ 0, \ 0), \ \underline{\hspace{0.5cm}} e0_data_type(d16s), \ \underline{\hspace{0.5cm}} e0_size(1, \ 1), \ \underline{\hspace{0.5cm}} ek_size(4, \ 1)), \ \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(3), \ \underline{\hspace{0.5cm}} identifier("S \leftarrow \ 1))$ VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_← type(d16s), e0 size(1, 1), ek size(MAX BLOCK WIDTH, MAX FILTER Y 7+1)), port(index(4), identifier("SVXY"), attributes(ACF ATTR VEC OUT STATIC FIXED), spatial dep(0, 0, 0, 0), ← e0 data type(d16s), e0 size(1, 1), ek size(MAX BLOCK WIDTH, MAX FILTER Y 7+1)), port(← $\underline{\quad} index(5), \ \underline{\quad} identifier("SVYY"), \ \underline{\quad} attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), \ \underline{\quad} spatial_dep(0, index(5), index(5),$ $0, \ 0, \ 0), \ \underline{\hspace{0.1cm}} = 0_data_type(d16s), \ \underline{\hspace{0.1cm}} = 0_size(1, \ 1), \ \underline{\hspace{0.1cm}} = k_size(MAX_BLOCK_WIDTH, \ MAX_FILTER_Y \leftarrow 1.00 + 1.$ _7+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MA← X_NMS_R-1, MAX_NMS_R+1)), __port(_index(7), __identifier("NMS"), __attributes(ACF_ATTR_VEC_O← UT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_ $BLOCK_WIDTH, MAX_NMS_R+1)), \\ __port(\\ __index(8), \\ __identifier("BXX"), \\ __attributes(ACF_ATTR_VEC \\ \\ \hookrightarrow \\ Order \\ Order$ _OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(M AX_BLOCK_WIDTH+MAX_FILTER_X_7-1, 1)), __port(__index(9), __identifier("BXY"), __attributes(A -- $CF_ATTR_VEC_OUT_STATIC_FIXED), \ \underline{\quad} spatial_dep(0,\ 0,\ 0), \ \underline{\quad} e0_data_type(d32s), \ \underline{\quad} e0_size(1,\ 0,\ 0), \ \underline{\quad} e0_size(1,\ 0,\ 0), \ \underline{\quad} e0_data_type(d32s), \ \underline{\quad} e0_d$ 1), __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_7-1, 1)), __port(__index(10), __identifier("BYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), e0 size(1, 1), ek size(MAX BLOCK WIDTH+MAX FILTER X 7-1, 1)))
- KERNEL INFO harris_corners_box5_nms5 (" harris_corners_box5_nms5 ", 11, __port(__index(0), _ ~ identifier("INPUT"), attributes(ACF ATTR VEC IN), spatial dep(2, 2, 1, 1), e0 data type(d08u), _e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("PARAMS"), __attributes(ACF_ATTR_← SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(7, $1)), \quad \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(2), \quad \underline{\hspace{0.5cm}} identifier("OUTPUT"), \quad \underline{\hspace{0.5cm}} attributes(ACF_ATTR_VEC_OUT), \quad \underline{\hspace{0.5cm}} spatial_dep(0, \underline{\hspace{0.5cm}} index(0, \underline{\hspace{0.5cm}} index(0,$ $0, \ 0, \ 0), \ \underline{\hspace{0.5cm}} e0_data_type(d16s), \ \underline{\hspace{0.5cm}} e0_size(1, \ 1), \ \underline{\hspace{0.5cm}} ek_size(4, \ 1)), \ \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(3), \ \underline{\hspace{0.5cm}} identifier("S \leftarrow \ 1))$ VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_5+1)), __port(__index(4), __identifier("SVXY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __ e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_5+1)), __port(← __index(5), __identifier("SVYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), $_$ e0_data_type(d16s), $_$ e0_size(1, 1), $_$ ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y \leftarrow _5+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MA X_NMS_R-1, MAX_NMS_R+1)), __port(__index(7), __identifier("NMS"), __attributes(ACF_ATTR_VEC_O UT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_← BLOCK WIDTH, MAX NMS R+1)), port(index(8), identifier("BXX"), attributes(ACF ATTR VEC _OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(M -AX_BLOCK_WIDTH+MAX_FILTER_X_5-1, 1)), __port(__index(9), __identifier("BXY"), __attributes(A← CF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_5-1, 1)), __port(__index(10), __identifier("BYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_5-1, 1)))
- KERNEL_INFO harris_corners_box3_nms5 (" harris_corners_box3_nms5 ", 11, __port(__index(0), __ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("PARAMS"), __attributes(ACF_ATTR_\circ SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(7, 1)), __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("S\circ VXX"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_\circ type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_3+1)), __port(__index(4), __identifier("SVXY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __\circ e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y_3+1)), __port(\circ index(5), __identifier("SVYY"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_FILTER_Y\circ 3+1)), __port(__index(6), __identifier("NMS_X"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH, MAX_BLOCK_WIDTH+MA\circ spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(MAX_BLOCK_WIDTH+MA\circ spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_siz

 $X_NMS_R-1, MAX_NMS_R+1)), _port(_index(7), _identifier("NMS"), _attributes(ACF_ATTR_VEC_O \leftarrow UT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(MAX_\leftarrow BLOCK_WIDTH, MAX_NMS_R+1)), _port(_index(8), _identifier("BXX"), _attributes(ACF_ATTR_VEC \leftarrow OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(M \leftarrow AX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)), _port(_index(9), _identifier("BXY"), _attributes(A \leftarrow CF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)), _port(_index(10), _identifier("BYY"), _attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)), _spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), _e0_size(1, 1), _ek_size(MAX_BLOCK_WIDTH+MAX_FILTER_X_3-1, 1)))$

- KERNEL_INFO harris_extract (" harris_extract ", 7, __port(_index(0), __identifier("COORD"), __ attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(MAX_CORNERS, 1)), __port(__index(1), __identifier("STREN"), __attributes(A\lefta CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(MAX_CORNERS, 1)), __port(__index(2), __identifier("COUNT"), __attributes(ACF_ATTR_SC\lefta L_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("COORD1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(MAX_CORNER_PER_CHU\lefta NK, 1)), __port(__index(4), __identifier("STREN1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(MAX_CORNER_PER_C\lefta HUNK, 1)), __port(__index(5), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(6), __identifier("PARAM\lefta S"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1), __ek_size(1, 1), __e0_size(1, 1
- KERNEL_INFO harris_sort_and_filter (" harris_sort_and_filter ", 7, __port(__index(0), __identifier("FE\leftrightarriangleriang

7.16.1 Detailed Description

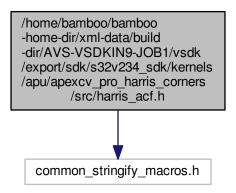
ACF metadata and wrapper function for the harris corner.

7.17 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/apexcv pro harris corners/src/harris acf.h File Reference

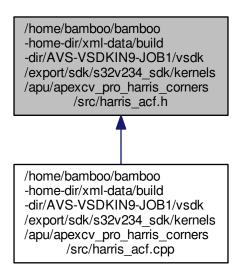
ACF metadata for the harris corner.

#include "common_stringify_macros.h"

Include dependency graph for harris_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define HARRIS_CORNERS_BOX7_NMS5_K harris_corners_box7_nms5
- #define HARRIS_CORNERS_BOX7_NMS5_KN XSTR(HARRIS_CORNERS_BOX7_NMS5_K)
- #define HARRIS CORNERS BOX5 NMS5 K harris corners box5 nms5
- #define HARRIS CORNERS BOX5 NMS5 KN XSTR(HARRIS CORNERS BOX5 NMS5 K)
- #define HARRIS CORNERS BOX3 NMS5 K harris corners box3 nms5
- #define HARRIS_CORNERS_BOX3_NMS5_KN XSTR(HARRIS_CORNERS_BOX3_NMS5_K)

- #define HARRIS_EXTRACT_K harris_extract
- #define HARRIS_EXTRACT_KN XSTR(HARRIS_EXTRACT_K)
- #define HARRIS_SORT_AND_FILTER_K harris_sort_and_filter
- #define HARRIS_SORT_AND_FILTER_KN XSTR(HARRIS_SORT_AND_FILTER_K)
- #define INPUT "INPUT"
- #define PARAMS "PARAMS"
- #define **OUTPUT** "OUTPUT"
- #define SVXX "SVXX"
- #define SVXY "SVXY"
- #define SVYY "SVYY"
- #define NMS X "NMS X"
- #define NMS "NMS"
- #define BXX "BXX"
- #define BXY "BXY"
- #define BYY "BYY"
- #define COORD "COORD"
- #define STREN "STREN"
- #define COUNT "COUNT"
- #define COORD1 "COORD1"
- #define STREN1 "STREN1"
- #define SRC "SRC"
- #define FEATURE "FEATURE"
- #define FEAT_OUT "FEAT_OUT"
- #define INDEX "INDEX"

Functions

- extKernelInfoDecl (harris_corners_box7_nms5)
- extKernelInfoDecI (harris_corners_box5_nms5)
- extKernelInfoDecl (harris_corners_box3_nms5)
- · extKernelInfoDecl (harris extract)
- extKernelInfoDecl (harris_sort_and_filter)

7.17.1 Detailed Description

ACF metadata for the harris corner.

7.18 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/apexcv_pro_hog/src/hog_apu.h File Reference

HOG APU kernel implementation.

7.18.1 Detailed Description

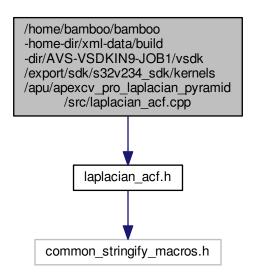
HOG APU kernel implementation.

7.19 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.cpp File

7.19 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.cpp File Reference

ACF metadata and wrapper function for the Laplacian Pyramid.

#include "laplacian_acf.h"
Include dependency graph for laplacian_acf.cpp:



Functions

• KERNEL_INFO horizontal_gaus_laplacian_mid (" horizontal_gaus_laplacian_mid ", 2, __port(__index(0), \leftarrow __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_V \leftarrow EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)))

KERNEL_INFO horizontal_gaus_laplacian_last (" horizontal_gaus_laplacian_last ", 2, __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_V \cong EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))

KERNEL_INFO vertical_gaus_laplacian_mid (" vertical_gaus_laplacian_mid ", 4, __port(__index(0), __
 identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("INPUT_GAUSS"), __attributes(ACF_A
 TTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)), __
 port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __
 __e0_data_type(d16s), __e0_size(1, 1), __ek_size(2, 2)), __port(__index(3), __identifier("OUTPUT_NEXT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 __ek_size(1, 1)))

ACF metadata for the 1x5 vertical Gaussian and Laplacian output mid level.

ACF metadata for the 5x1 horizontal Gaussian operation.

• KERNEL_INFO vertical_gaus_laplacian_last (" vertical_gaus_laplacian_last ", 4, __port(__index(0), __ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_GAUSS"), __attributes(ACF_A

```
\label{eq:total_total_dep} \begin{split} &TTR\_VEC\_IN), \\ &\_spatial\_dep(0, 0, 2, 2), \\ &\_e0\_data\_type(d16u), \\ &\_e0\_size(1, 1), \\ &\_ek\_size(1, 1)), \\ &\_ellowered_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_endering_ender
```

ACF metadata for the 1x5 vertical Gaussian and Laplacian output last level.

7.19.1 Detailed Description

ACF metadata and wrapper function for the Laplacian Pyramid.

7.19.2 Function Documentation

```
7.19.2.1 KERNEL_INFO horizontal_gaus_laplacian_mid ( "horizontal_gaus_laplacian_mid ", 2, __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)) )
```

ACF metadata for the 5x1 horizontal Gaussian operation.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.19.2.2 KERNEL_INFO vertical_gaus_laplacian_last ( " vertical_gaus_laplacian_last " , 4 , __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(_index(1), __identifier("INPUT_GAUSS"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(3), __identifier("OUTPUT_REC"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

ACF metadata for the 1x5 vertical Gaussian and Laplacian output last level.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.19.2.3 KERNEL_INFO vertical_gaus_laplacian_mid ( " vertical_gaus_laplacian_mid " , 4 , __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)) , __port(__index(1), __identifier("INPUT_GAUSS"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 2)) , __port(__index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(2, 2)) , __port(__index(3), __identifier("OUTPUT_NEXT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

ACF metadata for the 1x5 vertical Gaussian and Laplacian output mid level.

See also

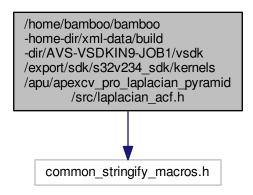
UG-10267-03 ACF User Guide, Section 3.2.2

7.20 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.h File

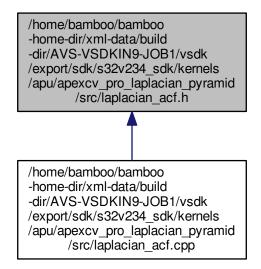
7.20 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_acf.h File Reference

ACF metadata for the Laplacian Pyramid.

#include "common_stringify_macros.h"
Include dependency graph for laplacian_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

• #define HORIZONTAL_GAUS_LAPLACIAN_MID_K horizontal_gaus_laplacian_mid

- #define HORIZONTAL_GAUS_LAPLACIAN_MID_KN XSTR(HORIZONTAL_GAUS_LAPLACIAN MID K)
- #define HORIZONTAL_GAUS_LAPLACIAN_LAST_K horizontal_gaus_laplacian_last
- #define HORIZONTAL_GAUS_LAPLACIAN_LAST_KN XSTR(HORIZONTAL_GAUS_LAPLACIAN_LAS← T K)
- #define VERTICAL GAUS LAPLACIAN MID K vertical gaus laplacian mid
- #define VERTICAL_GAUS_LAPLACIAN_MID_KN XSTR(VERTICAL_GAUS_LAPLACIAN_MID_K)
- #define VERTICAL_GAUS_LAPLACIAN_LAST_K vertical_gaus_laplacian_last
- #define VERTICAL_GAUS_LAPLACIAN_LAST_KN XSTR(VERTICAL_GAUS_LAPLACIAN_LAST_K)
- #define INPUT "INPUT"
- #define **OUTPUT** "OUTPUT"
- #define INPUT_GAUSS "INPUT_GAUSS"
- #define OUTPUT NEXT "OUTPUT NEXT"
- #define OUTPUT_REC "OUTPUT_REC"

Functions

- extKernelInfoDecI (horizontal_gaus_laplacian_mid)
- extKernelInfoDecl (horizontal gaus laplacian last)
- extKernelInfoDecl (vertical_gaus_laplacian_mid)
- extKernelInfoDecl (vertical_gaus_laplacian_last)

7.20.1 Detailed Description

ACF metadata for the Laplacian Pyramid.

7.21 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.cpp File Reference

APU Laplacian Pyramid Implementation.

7.21.1 Detailed Description

APU Laplacian Pyramid Implementation.

7.22 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.h File Reference

APU Laplacian Pyramid Header.

Functions

 void apu_pyr_horizontal_gaus_laplacian (const vec08u *lpvIn, int16u lStrideIn, vec16u *lpvOut, int16u l← StrideOut, int16u lChunkWidth, int16u lChunkHeight)

Apply a 5x1 horizontal gaussian filter to an image.

7.22 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/apexcv_pro_laplacian_pyramid/src/laplacian_apu.h File Reference

void apu_pyr_vertical_gaus_laplacian_mid (const vec08u *lpvIn, int16u lStrideIn, const vec16u *lpvIn_gauss, int16u lStrideIn_gauss, vec16s *lpvOut_lap, int16u lStrideOut_lap, vec08u *lpvOut_nex, int16u lStrideOut—nex, int16u lChunkWidth, int16u lChunkHeight)

Apply a 1x5 vertical gaussian filter to an image and generate laplacian and reduced outputs.

void apu_pyr_vertical_gaus_laplacian_last (const vec08u *lpvIn, int16u lStrideIn, const vec16u *lpvIn_gauss, int16u lStrideIn_gauss, vec16s *lpvOut_lap, int16u lStrideOut_lap, vec16s *lpvOut_out, int16u lStrideOut_cout, int16u lChunkWidth, int16u lChunkHeight)

Apply a 1x5 vertical gaussian filter to an image and generate laplacian and filtered outputs.

7.22.1 Detailed Description

APU Laplacian Pyramid Header.

image pyramid creation implementation for the APU.

7.22.2 Image Pyramid Creation

There are two common kinds of image pyramids: Gaussian pyramids and Laplacian pyramids. Here, we present the implementation of Laplacian pyramid creation.

To downsample an image (pyramid down), first the source image is convolved with a 5x5 Gaussian kernel, and then every even-numbered row and column is removed. As a result, the area is reduced to exactly one-quarter the area of the source image.

Laplacian output for each pyramid level is obtained by subtracting the convoluted image from the source image. The convoluted image is also used for the final output, which is necessary for the input image reconstruction.

7.22.3 Function Documentation

7.22.3.1 void apu_pyr_horizontal_gaus_laplacian (const vec08u * *lpvln*, int16u *lStrideIn*, vec16u * *lpvOut*, int16u *lStrideOut*, int16u *lChunkWidth*, int16u *lChunkHeight*)

Apply a 5x1 horizontal gaussian filter to an image.

5x1 horizontal gaussian filter is applied.

Parameters

lpvIn	Pointer to the source image. The source image is assumed to be padded according to the
	filter size. However, lpvIn points the top left corner of the <i>unpadded</i> image region.
lStrideIn	Stride of the padded source image.
lpvOut	Pointer to the destination image.
<i>IStrideOut</i>	Stride of the destination image.
<i>IChunkWidth</i>	Chunk width.
<i>IChunkHeight</i>	Chunk height.

7.22.3.2 void apu_pyr_vertical_gaus_laplacian_last (const vec08u * *lpvln*, int16u *IStrideIn*, const vec16u * *lpvln_gauss*, int16u *IStrideIn_gauss*, vec16s * *lpvOut_lap*, int16u *IStrideOut_lap*, vec16s * *lpvOut_out*, int16u *IStrideOut_out*, int16u *IChunkWidth*, int16u *IChunkHeight*)

Apply a 1x5 vertical gaussian filter to an image and generate laplacian and filtered outputs.

1x5 horizontal gaussian filter is applied. The filtered image is one of the outputs. Then the filered image is subtracted from de input image to generate the laplacian output.

Parameters

lpvIn	Pointer to the source image.
lStrideIn	Stride of the source image.
lpvln_gauss	Pointer to the horizontally filtered source image. The image is assumed to be padded ac-
	cording to the filter size. However, lpvIn points the top left corner of the <i>unpadded</i> image
	region.
IStrideIn_gauss	Stride of the padded filtered source image.
lpvOut_lap	Pointer to the destination image - laplacian pyramid.
IStrideOut_lap	Stride of the destination image - laplacian pyramid.
lpvOut_out	Pointer to the destination image - last level output.
IStrideOut_out	Stride of the destination image - last level output.
<i>IChunkWidth</i>	Chunk width.
<i>IChunkHeight</i>	Chunk height.

7.22.3.3 void apu_pyr_vertical_gaus_laplacian_mid (const vec08u * *IpvIn*, int16u *IStrideIn*, const vec16u * *IpvIn_gauss*, int16u *IStrideIn_gauss*, vec16s * *IpvOut_lap*, int16u *IStrideOut_lap*, vec08u * *IpvOut_nex*, int16u *IStrideOut_nex*, int16u *IChunkWidth*, int16u *IChunkHeight*)

Apply a 1x5 vertical gaussian filter to an image and generate laplacian and reduced outputs.

1x5 horizontal gaussian filter is applied. Then the filered image is subtracted from de input image to generate laplacian output and it is reduced to 1/2 to generate input for the next pyramid level.

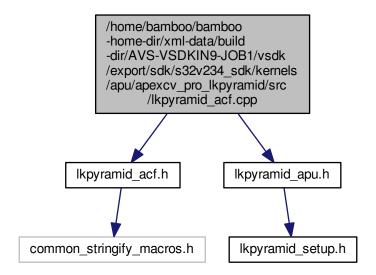
Parameters

lpvln	Pointer to the source image.
lStrideIn	Stride of the source image.
lpvln_gauss	Pointer to the horizontally filtered source image. The image is assumed to be padded ac-
	cording to the filter size. However, lpvIn points the top left corner of the <i>unpadded</i> image
	region.
lStrideIn_gauss	Stride of the padded filtered source image.
lpvOut_lap	Pointer to the destination image - laplacian pyramid.
IStrideOut_lap	Stride of the destination image - laplacian pyramid.
lpvOut_nex	Pointer to the destination image - input for next level.
IStrideOut_nex	Stride of the destination image - input for next level.
<i>IChunkWidth</i>	Chunk width.
<i>IChunkHeight</i>	Chunk height.

7.23 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.cpp File Reference

ACF metadata and wrapper function for the L-K optical flow pyramid.

#include "lkpyramid_acf.h"
#include "lkpyramid_apu.h"
Include dependency graph for lkpyramid acf.cpp:



Functions

- KERNEL_INFO lkpyramid_templ_bilinear_08u_7x7 (" lkpyramid_templ_bilinear_08u_7x7 ", 3, __port(_
 index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __
 e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 10)), __port(__index(1), __identifier("INPUT_1"), __
 attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXE
 D), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)))
 - ACF metadata for 7x7 bilinear interpolation.

ACF metadata for 7x7 X gradient using 3x3 centraldx/dy.

KERNEL_INFO lkpyramid_core_7x7 ("lkpyramid_core_7x7 ", 13, __port(_index(0), __identifier("INPUT ← _ 0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0 ← _ size(1, 1), __ek_size(20, 20)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VE ← C_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)), __ ← port(__index(2), __identifier("INPUT_2"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)), __port(__index(3), __identifier("INPUT_3"), ← __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)), __port(__index(4), __identifier("INPUT_4"), __attributes(ACF_ATTR_VEC_IN_FIXE ← D), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1), __port(__index(5), __identifier("INPUT_5"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data ← __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("INPUT_6"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data ← __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("INPUT_6"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data ← __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("INPUT_6"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data ← __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(6), __identifier("INPUT_6"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __attributes(ACF_ATTR_VEC_IN_FIXED), __identifier("INPUT_6"), __attributes(ACF_ATTR_VEC_IN_FIXED)

ACF metadata for 7x7 L-K pyramid core iteration.

7.23.1 Detailed Description

ACF metadata and wrapper function for the L-K optical flow pyramid.

.

7.23.2 Function Documentation

7.23.2.1 KERNEL_INFO lkpyramid_core_7x7 (" lkpyramid_core_7x7 " , 13 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(20, 20)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)) , __port(__index(2), __identifier("INPUT_2"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)), port(index(3), identifier("INPUT 3"), attributes(ACF ATTR VEC IN FIXED), spatial dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(__index(4), __identifier("INPUT_4"), _attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)) , __port(__index(5), __identifier("INPUT_5"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(6), __identifier("INPUT_6"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(8, 1)) , __port(__index(7), __identifier("INPUT_7"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)) , __port(__index(8), __identifier("OUTPUT_0"), _attributes(ACF_ATTR_VEC_OUT_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d32s), _e0_size(4, 1), _ek_size(1, 1)), __port(__index(9), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(10), __identifier("OUTPUT_2"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(11), __identifier("OUTPUT_3"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(8, 7)) , __port(__index(12), __identifier("OUTPUT_4"), _attributes(ACF_ATTR_VEC_OUT_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), __ek_size(8, 7)))

ACF metadata for 7x7 L-K pyramid core iteration.

**

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.23.2.2 KERNEL_INFO lkpyramid_ht_centraldxdy_7x7 ( " lkpyramid_ht_centraldxdy_7x7 " , 4 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)) , __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(8, 1)) , __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(__index(3), __e0_size(1, 1), __ek_size(8, 7)) )
```

ACF metadata for 7x7 X gradient using 3x3 centraldx/dy.

**

See also

UG-10267-03 ACF User Guide, Section 3.2.2

ACF metadata for 7x7 bilinear interpolation.

**

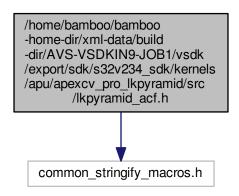
See also

UG-10267-03 ACF User Guide, Section 3.2.2

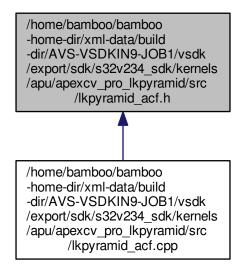
7.24 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_lkpyramid/src/lkpyramid_acf.h File Reference

ACF metadata for the L-K optical flow pyramid.

```
#include "common_stringify_macros.h"
Include dependency graph for lkpyramid_acf.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define LKPYRAMID_TEMPL_BILINEAR_08U_7X7_K | kpyramid_templ_bilinear_08u_7x7
- #define LKPYRAMID_HT_CENTRALDXDY_7X7_K |kpyramid_ht_centraldxdy_7x7
- #define LKPYRAMID HT CENTRALDXDY 7X7 KN XSTR(LKPYRAMID HT CENTRALDXDY 7X7 K)
- #define LKPYRAMID_CORE_7X7_K lkpyramid_core_7x7
- #define LKPYRAMID_CORE_7X7_KN XSTR(LKPYRAMID_CORE_7X7_K)
- #define INPUT_0 "INPUT_0"
- #define INPUT 1 "INPUT 1"
- #define OUTPUT 0 "OUTPUT 0"
- #define OUTPUT_1 "OUTPUT_1"
- #define INPUT_2 "INPUT_2"
- #define INPUT_3 "INPUT_3"
- #define INPUT 4 "INPUT 4"
- #define INPUT_5 "INPUT_5"
- #define INPUT_6 "INPUT_6"
- #define INPUT_7 "INPUT 7"
- #define OUTPUT_2 "OUTPUT_2"
- #define OUTPUT_3 "OUTPUT 3"
- #define OUTPUT_4 "OUTPUT_4"

Functions

- extKernelInfoDecl (lkpyramid templ bilinear 08u 7x7)
- extKernelInfoDecl (lkpyramid ht centraldxdy 7x7)
- extKernelInfoDecl (lkpyramid_core_7x7)

Reference 7.24.1 Detailed Description

241

ACF metadata for the L-K optical flow pyramid.

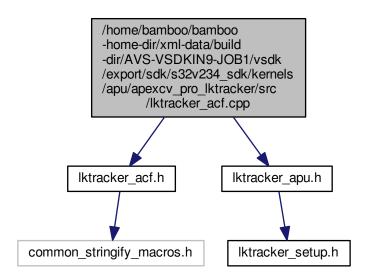
•

7.25 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_acf.cpp File Reference

ACF metadata and wrapper function for the L-K optical flow tracker.

```
#include "lktracker_acf.h"
#include "lktracker_apu.h"
```

Include dependency graph for lktracker_acf.cpp:



Functions

- KERNEL_INFO lktracker_templ_bilinear_08u_7x7 (" lktracker_templ_bilinear_08u_7x7 ", 3, __port(_ ← index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), ← __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 10)), __port(__index(1), __identifier("INPUT_1"), _ ← __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXE ← D), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)))
 - ACF metadata for 7x7 bilinear interpolation.
- KERNEL_INFO lktracker_ht_centraldxdy_7x7 (" lktracker_ht_centraldxdy_7x7 ", 4, __port(_index(0), _
 _ identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d08u), __e0_size(1, 1), __ek_size(12, 9)), __port(_index(1), __identifier("INPUT_1"), __attributes(A
 CF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1),
 __ek_size(12, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIX
 ED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)), __port(__index(3), __e0_size(1, 1), __ek_size(1, 1), __ek_

 $_$ identifier("OUTPUT_1"), $_$ attributes(ACF_ATTR_VEC_OUT_FIXED), $_$ spatial_dep(0, 0, 0, 0), $_$ e0 $_{\leftarrow}$ data_type(d16s), $_$ e0_size(1, 1), $_$ ek_size(8, 7)))

ACF metadata for 7x7 X gradient using 3x3 centraldx/dy.

 KERNEL_INFO lktracker_core_7x7 (" lktracker_core_7x7 ", 9, __port(__index(0), __identifier("INPUT_0"), ← __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), ek size(20, 20)), port(index(1), identifier("INPUT 1"), attributes(ACF ATTR VEC IN FIX↔ ED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)), __port(__index(2), __identifier("INPUT_2"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_ F_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)), __port(__index(4), __identifier("INPUT_4"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)), __port(__index(5), __identifier("INPUT← _5"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), _e0_size(1, 1), __ek_size(12, 1)), __port(__index(6), __identifier("OUTPUT_0"), __attributes(ACF_ATTR⊷ _VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)), _ \chi_ _port(__index(7), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, (0, 0, 0), e0 data type((08u), e0 size((1, 1)), ek size((8, 7)), port(index((8)), identifier("OUTP \leftarrow UT_2"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), _ ~ e0 size(1, 1), ek size(8, 7)))

ACF metadata for 7x7 L-K tracker core iteration.

7.25.1 Detailed Description

ACF metadata and wrapper function for the L-K optical flow tracker.

7.25.2 Function Documentation

7.25.2.1 KERNEL_INFO Iktracker_core_7x7 (" Iktracker_core_7x7 " , 9 , __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(20, 20)) , __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)) , __port(_index(2), __identifier("INPUT_2"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(_index(3), __identifier("INPUT_3"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(_index(4), __identifier("INPUT_4"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)) , __port(_index(5), __identifier("INPUT_5"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(12, 1)) , __port(_index(6), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)) , __port(_index(7), _identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1)) , __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1, 1), __ek_size(8, 7)) , __port(_index(8), _identifier("OUTPUT_2"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(_index(8), _identifier("OUTPUT_2"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)))

ACF metadata for 7x7 L-K tracker core iteration.

**

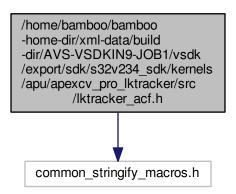
UG-10267-03 ACF User Guide, Section 3.2.2

```
7.25.2.2 KERNEL_INFO lktracker_ht_centraldxdy_7x7 ( " lktracker_ht_centraldxdy_7x7 " , 4 , __port(_index(0),
         __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0),
          __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)) , __port(__index(1), __identifier("INPUT_1"),
          __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1,
         1), __ek_size(12, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXED),
          __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(8, 7)) , __port(__index(3),
          __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s),
          __e0_size(1, 1), __ek_size(8, 7)) )
ACF metadata for 7x7 X gradient using 3x3 centraldx/dy.
See also
      UG-10267-03 ACF User Guide, Section 3.2.2
7.25.2.3 KERNEL_INFO lktracker_templ_bilinear_08u_7x7 ( " lktracker_templ_bilinear_08u_7x7 " , 3 ,
          __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0,
          0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 10)) , __port(__index(1), __identifier("INPUT_1"),
          __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(4, 1), __ek_size(1,
          1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0),
          __e0_data_type(d08u), __e0_size(1, 1), __ek_size(12, 9)) )
ACF metadata for 7x7 bilinear interpolation.
See also
      UG-10267-03 ACF User Guide, Section 3.2.2
```

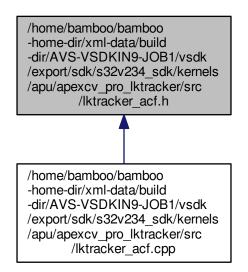
7.26 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_lktracker/src/lktracker_acf.h File Reference

ACF metadata for the L-K optical flow tracker.

#include "common_stringify_macros.h"
Include dependency graph for lktracker_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define LKTRACKER_TEMPL_BILINEAR_08U_7X7_K | ktracker_templ_bilinear_08u_7x7
- #define LKTRACKER_TEMPL_BILINEAR_08U_7X7_KN XSTR(LKTRACKER_TEMPL_BILINEAR_08U_ ← 7X7_K)
- #define LKTRACKER_HT_CENTRALDXDY_7X7_K lktracker_ht_centraldxdy_7x7
- #define LKTRACKER_HT_CENTRALDXDY_7X7_KN XSTR(LKTRACKER_HT_CENTRALDXDY_7X7_K)
- #define LKTRACKER_CORE_7X7_K lktracker_core_7x7

- #define LKTRACKER_CORE_7X7_KN XSTR(LKTRACKER_CORE_7X7_K)
- #define INPUT_0 "INPUT_0"
- #define INPUT_1 "INPUT_1"
- #define OUTPUT_0 "OUTPUT_0"
- #define OUTPUT_1 "OUTPUT_1"
- #define INPUT_2 "INPUT_2"
- #define INPUT 3 "INPUT 3"
- #define INPUT_4 "INPUT_4"
- #define INPUT 5 "INPUT 5"
- #define OUTPUT_2 "OUTPUT_2"

Functions

- extKernelInfoDecl (lktracker_templ_bilinear_08u_7x7)
- extKernelInfoDecl (lktracker ht centraldxdy 7x7)
- extKernelInfoDecl (lktracker_core_7x7)

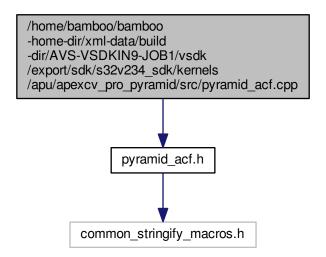
7.26.1 Detailed Description

ACF metadata for the L-K optical flow tracker.

7.27 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_acf.cpp File Reference

ACF metadata and wrapper function for the image pyramid creation.

```
#include "pyramid_acf.h"
Include dependency graph for pyramid_acf.cpp:
```



Functions

KERNEL_INFO horizontal_gaus (" horizontal_gaus ", 2, __port(__index(0), __identifier("INPUT"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek
 _size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))

ACF metadata for the 5x1 horizontal Gaussian operation.

KERNEL_INFO horizontal_gaus_and_expand (" horizontal_gaus_and_expand ", 2, __port(__index(0), _
 _ identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u),
 _ e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_V
 EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(2, 2)))

ACF metadata for the 5x1 horizontal Gaussian and expand operation.

KERNEL_INFO vertical_gaus ("vertical_gaus", 2, __port(__index(0), __identifier("INPUT"), __attributes(A CF_ATTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

ACF metadata for the 1x5 vertical Gaussian operation.

ACF metadata for the 1x5 vertical Gaussian and reduce operation.

7.27.1 Detailed Description

ACF metadata and wrapper function for the image pyramid creation.

7.27.2 Function Documentation

```
7.27.2.1 KERNEL_INFO horizontal_gaus ( "horizontal_gaus ", 2 , __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) , __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)) )
```

ACF metadata for the 5x1 horizontal Gaussian operation.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.27.2.2 KERNEL_INFO horizontal_gaus_and_expand ( "horizontal_gaus_and_expand ", 2, __port(_index(0), __identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(2, 2)) )
```

ACF metadata for the 5x1 horizontal Gaussian and expand operation.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.27.2.3 KERNEL_INFO vertical_gaus ( " vertical_gaus " , 2 , __port(__index(0), __identifier("INPUT"),
          _attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1,
         1)) , \_port(\_index(1), \_identifier("OUTPUT"), \_attributes(ACF_ATTR_VEC_OUT), \_spatial_dep(0, 0, 0, 0),
          __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

ACF metadata for the 1x5 vertical Gaussian operation.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

```
7.27.2.4 KERNEL_INFO vertical_gaus_and_reduce ( " vertical_gaus_and_reduce " , 2 , __port(__index(0), __identifier("INPUT"),
                                                                      __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 2, 2), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(2,
                                                                     2))\;,\;\; \_port(\_index(1),\; \_identifier("OUTPUT"),\; \_attributes(ACF\_ATTR\_VEC\_OUT),\; \_spatial\_dep(0,\;0,\;0,\;0),\; \_spatial\_dep(0,\;0,\;0),\; \_spatial\_dep
                                                                      __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)) )
```

ACF metadata for the 1x5 vertical Gaussian and reduce operation.

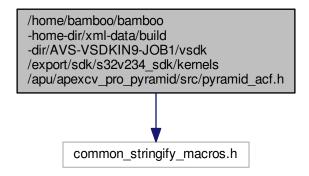
See also

UG-10267-03 ACF User Guide, Section 3.2.2

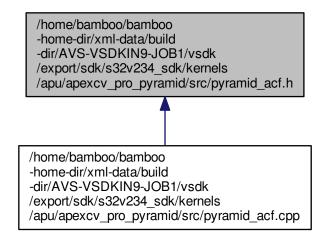
/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 7.28 _sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_acf.h File Reference

ACF metadata for the image pyramid creation.

```
#include "common_stringify_macros.h"
Include dependency graph for pyramid_acf.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define HORIZONTAL GAUS K horizontal gaus
- #define HORIZONTAL GAUS KN XSTR(HORIZONTAL GAUS K)
- #define HORIZONTAL_GAUS_AND_EXPAND_K horizontal_gaus_and_expand
- #define HORIZONTAL GAUS AND EXPAND KN XSTR(HORIZONTAL GAUS AND EXPAND K)
- #define VERTICAL GAUS K vertical gaus
- #define VERTICAL GAUS KN XSTR(VERTICAL GAUS K)
- #define VERTICAL_GAUS_AND_REDUCE_K vertical_gaus_and_reduce
- #define VERTICAL GAUS AND REDUCE KN XSTR(VERTICAL GAUS AND REDUCE K)
- #define INPUT "INPUT"
- #define **OUTPUT** "OUTPUT"

Functions

- extKernelInfoDecl (horizontal_gaus)
- extKernelInfoDecl (horizontal_gaus_and_expand)
- extKernelInfoDecl (vertical_gaus)
- extKernelInfoDecl (vertical_gaus_and_reduce)

7.28.1 Detailed Description

ACF metadata for the image pyramid creation.

7.29 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/apexcv_pro_pyramid/src/pyramid_apu.cpp File Reference

apu Image Pyramid Implementation.

Declaration of kernel functions for Remap.

7.31.2 Function Documentation

7.31.2.1 void remap_bilinear_grayscale (vec08u __cmem * dst, vec08u __cmem * src, vec16u __cmem * offset, vec08u __cmem * delta, int sstride, int dstride, int bw, int bh)

Performs remap of 8-bit grayscale pixel data.

Returns

None.

7.31.2.2 void remap_bilinear_rgb (vec32u __cmem * dst, vec32u __cmem * src, vec16u __cmem * offset, vec08u __cmem * delta, int sstride, int dstride, int bw, int bh)

Performs remap of RGBA pixel data..

Returns

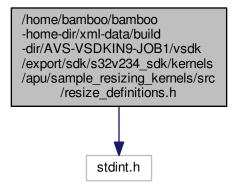
None.

7.32 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_resizing_kernels/src/resize_definitions.h File Reference

General definitions and declarations (Standard C), for Vertical Resizer.

```
#include <stdint.h>
```

Include dependency graph for resize_definitions.h:



Classes

• struct RESIZE DESCRIPTOR

Struct for Resize descriptor.

Macros

- #define RSZ_DECIMAL_SCL 16
- #define FLT BANK SIZE 32
- #define FLT_UP_SAMPLING
- #define FLT_DOWN_SAMPLING1

8 Phases, 4 Taps

• #define FLT_DOWN_SAMPLING2

4 Phases, 8 Taps

#define RESIZE_DESC_SIZE (POLYPHASE_FLT_SIZE*2 + (10*4))

Size of descriptors in Bytes.

7.32.1 Detailed Description

General definitions and declarations (Standard C), for Vertical Resizer.

7.32.2 Macro Definition Documentation

7.32.2.1 #define FLT_BANK_SIZE 32

Polyphase filter bank size

7.32.2.2 #define FLT_DOWN_SAMPLING1

Value:

```
{28, 200, 28, 0,\

15, 194, 47, 0,\
7, 178, 71, 0,\
2, 156, 98, 0,\
0, 128, 128, 0,\
0, 98, 156, 2,\
0, 71, 178, 7,\
0, 47, 194, 15}
```

8 Phases, 4 Taps

7.32.2.3 #define FLT_DOWN_SAMPLING2

Value:

```
{2, 32, 94, 94, 32, 2, 0, 0,\
0, 19, 80, 105, 47, 5, 0, 0,\
0, 11, 64, 106, 64, 11, 0, 0,\
0, 5, 47, 105, 80, 19, 0, 0}
```

4 Phases, 8 Taps

7.32.2.4 #define FLT_UP_SAMPLING

Value:

Filter banks 8 Phases, 4 Taps

7.32.2.5 #define RESIZE_DESC_SIZE (POLYPHASE_FLT_SIZE*2 + (10*4))

Size of descriptors in Bytes.

Did not use sizeof to prevent probable size mismatch of same types on different platforms sizeof(RESIZE_DESC←RIPTOR) in bytes

7.32.2.6 #define RSZ_DECIMAL_SCL 16

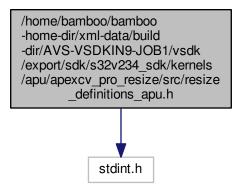
Fixed-point scaler

7.33 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/apexcv_pro_resize/src/resize_definitions_apu.h File Reference

General definitions and declarations (Standard C), for Vertical Resizer.

```
#include <stdint.h>
```

Include dependency graph for resize_definitions_apu.h:



Classes

struct RESIZE_DESCRIPTOR

Struct for Resize descriptor.

Macros

- #define RSZ DECIMAL SCL 16
- #define FLT_BANK_SIZE 32
- #define FLT_UP_SAMPLING
- #define FLT_DOWN_SAMPLING1

8 Phases, 4 Taps

• #define FLT_DOWN_SAMPLING2

4 Phases, 8 Taps

#define RESIZE_DESC_SIZE (POLYPHASE_FLT_SIZE*2 + (10*4))

Size of descriptors in Bytes.

7.33.1 Detailed Description

General definitions and declarations (Standard C), for Vertical Resizer.

7.33.2 Macro Definition Documentation

7.33.2.1 #define FLT_BANK_SIZE 32

Polyphase filter bank size

7.33.2.2 #define FLT_DOWN_SAMPLING1

Value:

```
{28, 200, 28, 0,\

15, 194, 47, 0,\
7, 178, 71, 0,\
2, 156, 98, 0,\
0, 128, 128, 0,\
0, 98, 156, 2,\
0, 71, 178, 7,\
0, 47, 194, 15}
```

8 Phases, 4 Taps

7.33.2.3 #define FLT_DOWN_SAMPLING2

Value:

```
{2, 32, 94, 94, 32, 2, 0, 0,\
0, 19, 80, 105, 47, 5, 0, 0,\
0, 11, 64, 106, 64, 11, 0, 0,\
0, 5, 47, 105, 80, 19, 0, 0}
```

4 Phases, 8 Taps

7.33.2.4 #define FLT_UP_SAMPLING

Value:

```
 \{0\ ,\ 256, \quad 0, \quad 0, \\ -11,\ 245,\ 23,\ -1, \\ -17,\ 220,\ 58,\ -5, \\ -18,\ 184,\ 100,\ -10, \\ -15,\ 143,\ 143,\ -15, \\ -10,\ 100,\ 184,\ -18, \\ -5,\ 58,\ 220,\ -17, \\ -1,\ 23,\ 245,\ -11\}
```

Filter banks 8 Phases, 4 Taps

7.33.2.5 #define RESIZE_DESC_SIZE (POLYPHASE_FLT_SIZE*2 + (10*4))

Size of descriptors in Bytes.

Did not use sizeof to prevent probable size mismatch of same types on different platforms sizeof(RESIZE_DESC←RIPTOR) in bytes

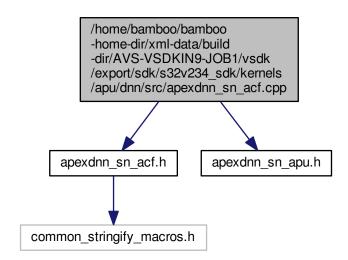
7.33.2.6 #define RSZ_DECIMAL_SCL 16

Fixed-point scaler

7.34 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/dnn/src/apexdnn_sn_acf.cpp File Reference

ACF metadata and wrapper function for the Convolutional Neural Networks.

```
#include "apexdnn_sn_acf.h"
#include "apexdnn_sn_apu.h"
Include dependency graph for apexdnn_sn_acf.cpp:
```



Functions

• KERNEL_INFO apu_conv3x3mps1_module_nhcw_forward (" apu_conv3x3mps1_module_nhcw_forward ", 7, __port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_ \circ dep(0, 0, 1, 1), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 4)), __port(__index(1), __identifier("I\circ NPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_\circ data_type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(2), __identifier("INPUT_PARAMS"), _\circ attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0\circ size(1, 1), __ek_size(32, 1)), __port(__index(3), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(A\circ CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(12, 1)), __port(__index(4), __identifier("OUTPUT_CONV3X3"), __attributes(ACF_ATTR_VEC_\circ OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(5), __identifier("OUTPUT_MP"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _\circ spatial_dep(0, 0, 0, 0), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6), _\circ attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_\circ data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)))

ACF metadata for APEXDNN library.

• KERNEL_INFO apu_e1e3s1_module_nhcw_forward (" apu_e1e3s1_module_nhcw_forward ", 7, __port(\(\) __index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(_index(1), __identifier("INPUT_\(\) WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_\(\) type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(_index(2), __identifier("INPUT_PARAMS"), __\(\) attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0\(\) _size(1, 1), __ek_size(32, 1)), __port(__index(3), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(A\(\) CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1),

- __ek_size(12, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_\circ
 STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), \circ
 _port(__index(5), __identifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __\circ
 spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6), __\circ
 identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_\circ
 data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)))
- KERNEL INFO apu e1e3mps1 module nhcw forward (" apu e1e3mps1 module nhcw forward ", 9, $\underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(0), \ \underline{\hspace{0.5cm}} identifier("INPUT_IMAGE"), \ \underline{\hspace{0.5cm}} attributes(ACF_ATTR_SCL_IN_FIXED), \ \underline{\hspace{0.5cm}} spatial_{\leftarrow}$ dep(0, 0, 1, 1), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("I← NPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(2), __identifier("INPUT_PARAMS"), _ <-_attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0← _size(1, 1), __ek_size(32, 1)), __port(_index(3), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(A CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, _ek_size(12, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_ OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(5), __identifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), spatial dep(0, 0, 0, 0), e0 data type(d08s), e0 size(1, 1), ek size(2, 1)), port(d08s), \leftrightarrow identifier("OUTPUT MP1"), attributes(ACF ATTR VEC OUT STATIC FIXED), spatial dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(7), __identifier("OUTPUT_MP3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), _e0_size(1, 1), __ek_size(2, 1)), __port(__index(8), __identifier("OUTPUT_S1"), __attributes(ACF_AT↔ TR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2,
- KERNEL INFO apu e1e3mp module nhcw forward (" apu e1e3mp module nhcw forward ", 8, \leftarrow port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, $0, \ 1, \ 1), \ __e0_data_type(d08s), \ __e0_size(1, \ 1), \ __ek_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ __identifier("INPU\leftarrow 1, \ 1), \ __ev_size(2, \ 4)), \ __port(__index(1), \ 2), \ -_port(__index(1), \ 2), \ -_port$ T_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data↔ _type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(2), __identifier("INPUT_PARAMS"), __ attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_← CF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), _ek_size(12, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT \Leftarrow STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), \leftarrow __port(__index(5), __identifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __ spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6), __ identifier("OUTPUT_MP1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0↔ data type(d08s), e0 size(1, 1), ek size(2, 1)), port(index(7), identifier("OUTPUT MP3"), \leftarrow attributes(ACF ATTR VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)))
- KERNEL_INFO apu_e1e3_module_nhcw_forward (" apu_e1e3_module_nhcw_forward ", 6, __port(_ ← index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(1), __identifier("INPUT_WEIGH←T"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(AC←F_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), ← ek_size(32, 1)), __port(__index(3), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(ACF_ATTR←SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(12, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), ← spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(5), _← identifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)))
- KERNEL_INFO apu_eltmulcred_module_nhcw_forward (" apu_eltmulcred_module_nhcw_forward ", 6, __port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial -- __dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __-identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("INPUT_BIAS"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1)

```
1), __ek_size(1, 1)), __port(__index(3), __identifier("INPUT_PARAMS"), __attributes(ACF_ATTR_SCL
    _IN_STATIC_FIXED),    __spatial_dep(0, 0, 0, 0),    __e0_data_type(d32s),    __e0_size(1, 1),    _ ek size(32,
   1)), \_\_port(\_\_index(4), \_\_identifier("OUTPUT\_SCRATCH\_BUF"), \_\_attributes(ACF\_ATTR\_SCL\_OUT \leftarrow Continuous Conti
   _STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)),
    __port(__index(5), __identifier("OUTPUT"), __attributes(ACF_ATTR_SCL_OUT_FIXED), __spatial_dep(0,
   0, 0, 0), e0 data type(d08s), e0 size(1, 1), ek size(1, 1)))
• KERNEL INFO apu e1ap module nhcw forward (" apu e1ap module nhcw forward ", 7, port( -
    \underline{\text{dep}(0,\ 0,\ 0,\ 0)},\ \underline{\text{e0}}\underline{\text{data}}\underline{\text{type}(\text{d08s})},\ \underline{\text{e0}}\underline{\text{size}(1,\ 1)},\ \underline{\text{ek}}\underline{\text{size}(2,\ 1)}),\ \underline{\text{port}}\underline{\text{(index}(1),\ \underline{\text{--}}}\underline{\text{constant}}
   identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __~
   e0_data_type(d08s), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(2), __identifier("INPUT_ACCM"),
      _attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1,
   1), \_ek_size(1, 1)), \_port(\_index(3), \_identifier("INPUT_PARAMS"), \_attributes(ACF_ATTR_SCL\leftrightarrow
    1)), \_port(\_index(4), \_identifier("OUTPUT_SCRATCH_BUF"), \_attributes(ACF_ATTR_SCL_OUT_\hookleftarrow
   STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(12, 1)),
    __port(__index(5), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), \Leftarrow
    __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6),
      _identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_←
   data type(d16s), e0 size(1, 1), ek size(1, 1))
```

7.34.1 Detailed Description

ACF metadata and wrapper function for the Convolutional Neural Networks.

.

7.34.2 Function Documentation

```
7.34.2.1 KERNEL_INFO apu_conv3x3mps1_module_nhcw_forward ( "apu_conv3x3mps1_module_nhcw_forward ", 7 , __port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 4)) , __port(__index(1), __identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(64, 1)) , __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(32, 1)) , __port(__index(3), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(12, 1)) , __port(__index(4), __identifier("OUTPUT_CONV3X3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)) , __port(__index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)) , __port(__index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(2, 1)) )
```

ACF metadata for APEXDNN library.

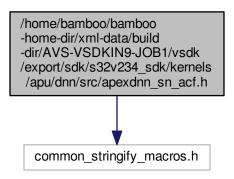
See also

UG-10267-03 ACF User Guide, Section 3.2.2

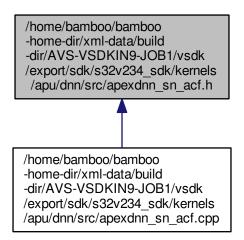
7.35 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/dnn/src/apexdnn_sn_acf.h File Reference

ACF metadata and wrapper function for the APEXDNN Library.

#include "common_stringify_macros.h" Include dependency graph for apexdnn_sn_acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- APU CONV3X3MPS1 MODULE NHCW FORWARD K apu conv3x3mps1 module nhcw \(-- \) #define forward
- #define APU CONV3X3MPS1 MODULE NHCW FORWARD KN XSTR(APU CONV3X3MPS1 MODU← LE_NHCW_FORWARD_K)
- #define APU E1E3S1 MODULE NHCW FORWARD K apu e1e3s1 module nhcw forward
- #define APU_E1E3S1_MODULE_NHCW_FORWARD_KN XSTR(APU_E1E3S1_MODULE_NHCW_FOR WARD K)
- #define APU_E1E3MPS1_MODULE_NHCW_FORWARD_K apu_e1e3mps1_module_nhcw_forward

- #define APU E1E3MP MODULE NHCW FORWARD K apu e1e3mp module nhcw forward
- #define APU_E1E3MP_MODULE_NHCW_FORWARD_KN XSTR(APU_E1E3MP_MODULE_NHCW_FO
 RWARD K)
- #define APU E1E3 MODULE NHCW FORWARD K apu e1e3 module nhcw forward
- #define APU_ELTMULCRED_MODULE_NHCW_FORWARD_K apu_eltmulcred_module_nhcw_forward
- #define APU_ELTMULCRED_MODULE_NHCW_FORWARD_KN XSTR(APU_ELTMULCRED_MODULE →
 _NHCW_FORWARD_K)
- #define APU_E1AP_MODULE_NHCW_FORWARD_K apu_e1ap_module_nhcw_forward
- #define APU_E1AP_MODULE_NHCW_FORWARD_KN XSTR(APU_E1AP_MODULE_NHCW_FORWA↔ RD K)
- #define INPUT_IMAGE "INPUT_IMAGE"
- #define INPUT_WEIGHT "INPUT_WEIGHT"
- #define INPUT_PARAMS "INPUT_PARAMS"
- #define OUTPUT SCRATCH BUF "OUTPUT SCRATCH BUF"
- #define OUTPUT_CONV3X3 "OUTPUT_CONV3X3"
- #define OUTPUT MP "OUTPUT MP"
- #define OUTPUT_S1 "OUTPUT S1"
- #define OUTPUT_E1 "OUTPUT_E1"
- #define OUTPUT_E3 "OUTPUT_E3"
- #define OUTPUT_MP1 "OUTPUT MP1"
- #define OUTPUT MP3 "OUTPUT MP3"
- #define INPUT_BIAS "INPUT_BIAS"
- #define **OUTPUT** "OUTPUT"
- #define INPUT_ACCM "INPUT_ACCM"

Functions

- extKernelInfoDecl (apu conv3x3mps1 module nhcw forward)
- extKernelInfoDecI (apu_e1e3s1_module_nhcw_forward)
- extKernelInfoDecI (apu_e1e3mps1_module_nhcw_forward)
- extKernelInfoDecI (apu_e1e3mp_module_nhcw_forward)
- extKernelInfoDecl (apu_e1e3_module_nhcw_forward)
- extKernelInfoDecI (apu_eltmulcred_module_nhcw_forward)
- extKernelInfoDecl (apu_e1ap_module_nhcw_forward)

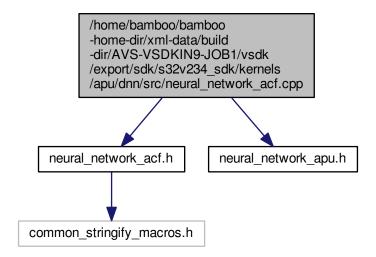
7.35.1 Detailed Description

ACF metadata and wrapper function for the APEXDNN Library.

7.36 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/dnn/src/neural network acf.cpp File Reference

ACF metadata and wrapper function for the Convolutional Neural Networks.

```
#include "neural_network_acf.h"
#include "neural_network_apu.h"
Include dependency graph for neural network acf.cpp:
```



Functions

 KERNEL INFO apu fire2 conv1mps1 forward (" apu fire2 conv1mps1 forward ", 8, port(index(0), ← identifier("INPUT IMAGE"), attributes(ACF ATTR SCL IN FIXED), spatial dep(0, 0, 2, 3), e0↔ _data_type(d08s), __e0_size(1, 1), __ek_size(700, 12)), __port(__index(1), __identifier("INPUT_WEIGH⊷ T"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), _e0_size(1, 1), __ek_size(64, 7)), __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(ACF↩ $_ATTR_SCL_IN_STATIC_FIXED), \ __spatial_dep(0,\ 0,\ 0,\ 0),\ __e0_data_type(d08s),\ __e0_size(1,\ 1),\ _\hookleftarrow (1,\ 1),\ _$ _ek_size(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_ OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(112, 1)), port(index(4), identifier("OUTPUT SCRATCH BUF"), attributes(ACF ATTR SCL OUT S↔ TATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d32s), e0 size(1, 1), ek size(9, 1)), \leftarrow port(_index(5), _identifier("OUTPUT_CONV1"), _attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(222, 3)), __port(__index(6), _<-_identifier("OUTPUT_MP"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(112, 1)), __port(__index(7), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), ek size(56, 3)))

ACF metadata for Convolutional Neural Networks.

- KERNEL_INFO apu_fire4_e1e3s1_forward (" apu_fire4_e1e3s1_forward ", 7, __port(__index(0), __ identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0_ data_type(d08s), __e0_size(1, 1), __ek_size(912, 5)), __port(__index(1), __identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __ e0_size(1, 1), __ek_size(64, 5)), __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(ACF_Adexize(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_OUdexize(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_OUdexize(16, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __didentifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __didentifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 1)), __port(__index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 1)), __port(__index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 1)), __port(__index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 5)))
- KERNEL INFO apu fire5_e1e3mps1_forward (" apu fire5_e1e3mps1_forward ", 10, port(index(0), identifier("INPUT IMAGE"), attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0 <-_data_type(d08s), __e0_size(1, 1), __ek_size(1808, 4)), __port(__index(1), __identifier("INPUT_WEIGH T"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __ek_size(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC← _OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(56, 1)), __port(__index(4), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(ACF_ATTR_SCL_OUT_S↔ TATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(9, 1)), _ <-_port(__index(5), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __ $spatial_dep(0,\ 0,\ 0,\ 0),\ __e0_data_type(d08s),\ __e0_size(1,\ 1),\ __ek_size(112,\ 3)),\ __port(__index(6),\ \hookleftarrow 1,\ 0)$ _identifier("OUTPUT_E3"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(112, 3)), __port(__index(7), __identifier("OUTP← UT_MP1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data ← type(d08s), e0 size(1, 1), ek size(56, 1)), port(index(8), identifier("OUTPUT MP3"), \leftarrow attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), _ ← _e0_size(1, 1), __ek_size(56, 1)), __port(__index(9), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR↔ _VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(28, 2)))
- KERNEL_INFO apu_fire7_e1e3s1_forward (" apu_fire7_e1e3s1_forward ", 7, __port(__index(0), __ identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 1, 1), __e0_ data_type(d08s), __e0_size(1, 1), __ek_size(1360, 3)), __port(__index(1), __identifier("INPUT_WEIGHT"),

- __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(64, 31)), __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_OU_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(28, 1)), __port(__index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(84, 1)), __port(__index(5), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(84, 1)), __port(_index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(84, 1)), __port(_index(6), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(28, 3)))

- KERNEL_INFO apu_fire9_2nd_s1_forward (" apu_fire9_2nd_s1_forward ", 5, __port(__index(0), __ \iff identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_ \iff data_type(d08s), __e0_size(1, 1), __ek_size(7184, 1)), __port(__index(1), __identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __ \iff e0_size(1, 1), __ek_size(64, 9)), __port(__index(2), __identifier("INPUT_PARAMS"), __attributes(ACF_A \iff TTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_ \iff size(16, 1)), __port(__index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_OUT_ \iff STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(14, 1)), __ \iff port(__index(4), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(14, 1)))

 $port(_index(4), __identifier("OUTPUT_E1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 2)), __port(__index(5), __identifier("OUT \leftarrow PUT_E3"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(56, 2)))$

- KERNEL_INFO apu_fire10_2nd_conv10ap_top_forward ("apu_fire10_2nd_conv10ap_top_forward ", 7, \(\toport(_index(0), _identifier("INPUT_IMAGE"), _attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), _\(\toport(_index(0), _0, 0), _e0_data_type(d08s), _e0_size(1, 1), _ek_size(7168, 6)), _port(_index(1), _identifier("INPUT_WEIGHT"), _attributes(ACF_ATTR_VEC_IN_FIXED), _spatial_dep(0, 0, 0, 0), _\(\toportupe e0_data_type(d08s), _e0_size(1, 1), _ek_size(64, 9)), _port(_index(2), _identifier("INPUT_ACCM"), _attributes(ACF_ATTR_VEC_IN_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(1, 1)), _port(_index(3), _identifier("INPUT_PARAMS"), _attributes(ACF_ATTR_SCL_IN\(\toport(\) STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08s), _e0_size(1, 1), _ek_size(16, 1)), _\(\toport(\) port(_index(4), _identifier("OUTPUT_ROW_BUF"), _attributes(ACF_ATTR_VEC_OUT_STATIC_FIX\(\toportupe ED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d32s), _e0_size(1, 1), _ek_size(14, 1)), _port(_index(5), _identifier("OUTPUT_CONV10"), _attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d08s), _e0_size(1, 1), _ek_size(14, 1)), _port(_index(6), _identifier("OU\(\toportupe TPUT_AP"), _attributes(ACF_ATTR_VEC_OUT_FIXED), _spatial_dep(0, 0, 0, 0), _e0_data_type(d16s), _e0_size(1, 1), _ek_size(1, 1)))
- KERNEL_INFO apu_fire10_2nd_conv10ap_bottom_forward (" apu_fire10_2nd_conv10ap_bottom_ convard ", 7, __port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_STATIC convard ", 7, __port(__index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_STATIC convard convard convariant convari

7.36.1 Detailed Description

ACF metadata and wrapper function for the Convolutional Neural Networks.

7.36.2 Function Documentation

```
7.36.2.1 KERNEL_INFO apu_fire2_conv1mps1_forward ( "apu_fire2_conv1mps1_forward ", 8 , __port(_index(0), __identifier("INPUT_IMAGE"), __attributes(ACF_ATTR_SCL_IN_FIXED), __spatial_dep(0, 0, 2, 3), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(700, 12)) , __port(_index(1), __identifier("INPUT_WEIGHT"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(64, 7)) , __port(_index(2), __identifier("INPUT_PARAMS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(16, 1)) , __port(_index(3), __identifier("OUTPUT_ROW_BUF"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(112, 1)) , __port(_index(4), __identifier("OUTPUT_SCRATCH_BUF"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(9, 1)) , __port(_index(5), __identifier("OUTPUT_CONV1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(222, 3)) , __port(_index(6), __identifier("OUTPUT_MP"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(112, 1)) , __port(_index(7), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(112, 1)) , __port(_index(7), __identifier("OUTPUT_S1"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_size(1, 1), __ek_size(56, 3)) )
```

ACF metadata for Convolutional Neural Networks.

**

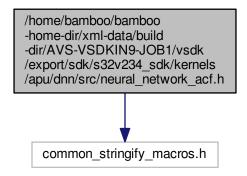
See also

UG-10267-03 ACF User Guide, Section 3.2.2

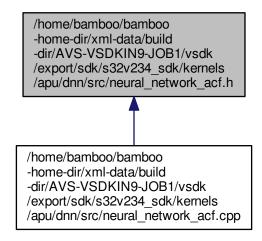
7.37 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/dnn/src/neural_network_acf.h File Reference

ACF metadata and wrapper function for the Convolutional Neural Networks.

#include "common_stringify_macros.h"
Include dependency graph for neural network acf.h:



This graph shows which files directly or indirectly include this file:



Macros

- #define APU_FIRE2_CONV1MPS1_FORWARD_K apu_fire2_conv1mps1_forward
- #define APU_FIRE2_CONV1MPS1_FORWARD_KN XSTR(APU_FIRE2_CONV1MPS1_FORWARD_K)
- #define APU_FIRE3_E1E3S1_FORWARD_K apu_fire3_e1e3s1_forward
- #define APU_FIRE3_E1E3S1_FORWARD_KN XSTR(APU_FIRE3_E1E3S1_FORWARD_K)
- #define APU_FIRE4_E1E3S1_FORWARD_K apu_fire4_e1e3s1_forward
- #define APU_FIRE4_E1E3S1_FORWARD_KN XSTR(APU_FIRE4_E1E3S1_FORWARD_K)
- #define APU_FIRE5_E1E3MPS1_FORWARD_K apu_fire5_e1e3mps1_forward
- #define APU_FIRE5_E1E3MPS1_FORWARD_KN XSTR(APU_FIRE5_E1E3MPS1_FORWARD_K)
- #define APU_FIRE6_E1E3S1_FORWARD_K apu_fire6_e1e3s1_forward
- #define APU_FIRE6_E1E3S1_FORWARD_KN XSTR(APU_FIRE6_E1E3S1_FORWARD_K)
- #define APU_FIRE7_E1E3S1_FORWARD_K apu_fire7_e1e3s1_forward
- #define APU_FIRE7_E1E3S1_FORWARD_KN XSTR(APU_FIRE7_E1E3S1_FORWARD_K)
- #define APU_FIRE8_E1E3S1_FORWARD_K apu_fire8_e1e3s1_forward
- #define APU_FIRE8_E1E3S1_FORWARD_KN XSTR(APU_FIRE8_E1E3S1_FORWARD_K)
- #define APU_FIRE9_1ST_E1E3MP_FORWARD_K apu_fire9_1st_e1e3mp_forward
- #define APU_FIRE9_1ST_E1E3MP_FORWARD_KN XSTR(APU_FIRE9_1ST_E1E3MP_FORWARD_K)
- #define APU_FIRE9_2ND_S1_FORWARD_K apu_fire9_2nd_s1_forward
- #define APU_FIRE9_2ND_S1_FORWARD_KN XSTR(APU_FIRE9_2ND_S1_FORWARD_K)
- #define APU_FIRE10_1ST_E1E3_FORWARD_K apu_fire10_1st_e1e3_forward
- #define APU_FIRE10_1ST_E1E3_FORWARD_KN XSTR(APU_FIRE10_1ST_E1E3_FORWARD_K)
- #define APU_FIRE10_2ND_CONV10AP_TOP_FORWARD_K apu_fire10_2nd_conv10ap_top_forward
- #define APU_FIRE10_2ND_CONV10AP_TOP_FORWARD_KN XSTR(APU_FIRE10_2ND_CONV10AP_

 TOP FORWARD K)
- #define APU_FIRE10_2ND_CONV10AP_BOTTOM_FORWARD_K apu_fire10_2nd_conv10ap_bottom_← forward
- #define INPUT IMAGE "INPUT IMAGE"
- #define INPUT_WEIGHT "INPUT_WEIGHT"

- #define INPUT_PARAMS "INPUT_PARAMS"
- #define OUTPUT_ROW_BUF "OUTPUT_ROW_BUF"
- #define OUTPUT_SCRATCH_BUF "OUTPUT_SCRATCH_BUF"
- #define OUTPUT_CONV1 "OUTPUT_CONV1"
- #define OUTPUT_MP "OUTPUT_MP"
- #define OUTPUT_S1 "OUTPUT_S1"
- #define OUTPUT_E1 "OUTPUT_E1"
- #define OUTPUT_E3 "OUTPUT_E3"
- #define OUTPUT_MP1 "OUTPUT_MP1"
- #define OUTPUT_MP3 "OUTPUT MP3"
- #define INPUT_ACCM "INPUT_ACCM"
- #define OUTPUT_CONV10 "OUTPUT CONV10"
- #define **OUTPUT AP** "OUTPUT AP"

Functions

- extKernelInfoDecl (apu_fire2_conv1mps1_forward)
- extKernelInfoDecl (apu fire3 e1e3s1 forward)
- extKernelInfoDecl (apu_fire4_e1e3s1_forward)
- extKernelInfoDecl (apu fire5 e1e3mps1 forward)
- extKernelInfoDecl (apu_fire6_e1e3s1_forward)
- extKernelInfoDecl (apu fire7 e1e3s1 forward)
- extKernelInfoDecl (apu_fire8_e1e3s1_forward)
- extKernelInfoDecl (apu fire9 1st e1e3mp forward)
- extKernelInfoDecl (apu fire9 2nd s1 forward)
- extKernelInfoDecl (apu_fire10_1st_e1e3_forward)
- extKernelInfoDecl (apu fire10 2nd conv10ap top forward)
- extKernelInfoDecI (apu_fire10_2nd_conv10ap_bottom_forward)

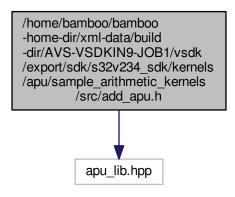
7.37.1 Detailed Description

ACF metadata and wrapper function for the Convolutional Neural Networks.

/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 7.38 _sdk/kernels/apu/sample_arithmetic_kernels/src/add_apu.h File Reference

element-wise addition implementation for APEX

#include "apu_lib.hpp"
Include dependency graph for add apu.h:



Functions

- void add (vec16u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)
 Elementwise unsigned 8bit addition => unsigned 16bit.
- void add_in16s_out32s (vec32s *dst, vec16s *srcImage0, vec16s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 16bit addition => signed 32bit.

void add_in32s_out32s (vec32s *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit addition => signed 32bit.

void add_in32u_out32u (vec32u *dst, vec32u *srcImage0, vec32u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 32bit addition => unsigned 32bit.

void add_in64s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcImage0_high, vec32u *srcImage0←
 _low, vec32s *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit addition => signed 64bit.

void add_in64u_out64u (vec32u *dst_high, vec32u *dst_low, vec32u *srcImage0_high, vec32u *src
 Image0_low, vec32u *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 64bit addition => unsigned 64bit.

void add_in32Q3_28_out32Q3_28 (vec32s *dstInt, vec32s *dstFrac, vec32s *srcImage0, vec32s *src
Image1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise 32bit Q3.28 addition => 64bit in [Q3.28] (integer part) and {Q3.28} (fractional part) format.

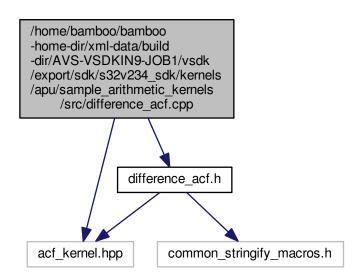
7.38.1 Detailed Description

element-wise addition implementation for APEX

7.39 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ sdk/kernels/apu/sample_arithmetic_kernels/src/difference_acf.cpp File

7.39 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample arithmetic kernels/src/difference acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "difference_acf.h"
Include dependency graph for difference_acf.cpp:
```



Functions

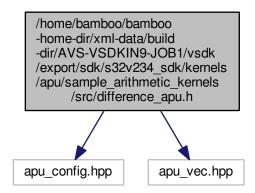
- KERNEL_INFO apu_diff_in08u_out16s (" apu_diff_in08u_out16s ", 3, __port(__index(0), __identifier("I← NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0← _size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC← _IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0← data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in16s_out16s (" apu_diff_in16s_out16s ", 3, __port(_index(0), __identifier("I↔ NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0↔ __size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC↔ __IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ↔ index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_↔ data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in16s_out32s (" apu_diff_in16s_out32s ", 3, __port(__index(0), __identifier("I← NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC← IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_diff_in32s_out32s (" apu_diff_in32s_out32s ", 3, __port(__index(0), __identifier("I← NPUT_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0← _size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC← _IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))

KERNEL_INFO apu_diff_in64s_out64s (" apu_diff_in64s_out64s ", 6, __port(__index(0), __identifier("IN← PUT_A_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), _← __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_A_LOW"), __attributes(ACF_A← TTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __oport(__index(2), __identifier("INPUT_B_HIGH"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("INPUT_B_← LOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("OUTPUT_HIGH"), __attributes(ACF_ATTR_VEC_OU← T), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("OUTPUT_LOW"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

7.40 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/difference_apu.h File Reference

Computes the pixelwise difference btw. two images.

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
Include dependency graph for difference apu.h:
```



Functions

void difference_filter_in08u_out16s (vec16s *dst, vec08u *srcA, vec08u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 8bit images => signed 16bit.

void difference_filter_in16s_out16s (vec16s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 16bit.

void difference_filter_in16s_out32s (vec32s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

void difference_filter_in32s_out32s (vec32s *dst, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 16bit images => signed 32bit.

7.41 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_acf.cpp File Reference

• void difference_filter_in32u_out32s (vec32s *dst, vec32u *srcA, vec32u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two unsigned 32bit images => signed 32bit.

• void difference_filter_in32s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Pixel-wise difference between two signed 32bit images => signed 64bit.

void difference_filter_in64s_out64s (vec32s *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStride
 Width)

Pixel-wise difference between two signed 64bit images => signed 64bit.

void difference_filter_in64u_out64s (vec32s *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStride Width)

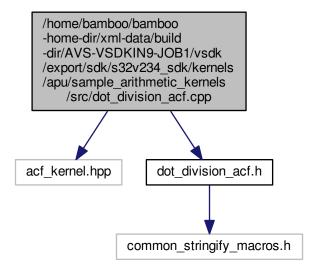
Pixel-wise difference between two unsigned 64bit images => signed 64bit.

7.40.1 Detailed Description

Computes the pixelwise difference btw. two images.

7.41 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_division_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "dot_division_acf.h"
Include dependency graph for dot division acf.cpp:
```



Functions

• KERNEL_INFO apu_dot_division (" apu_dot_division ", 3, __port(__index(0), __identifier("INPUT_A"), __
attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_
size(1, 1)), __port(__index(1), __identifier("INPUT_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPU
T"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))

- KERNEL_INFO apu_dot_log2 (" apu_dot_log2 ", 2, __port(_index(0), __identifier("Log2_input"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek
 _size(1, 1)), __port(_index(1), __identifier("Log2_fact"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_inv_NewtonRaphson ("apu_dot_inv_NewtonRaphson ", 4, __port(__index(0), __
 identifier("Inv_Inverse_divisor"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data
 _type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("Inv_divisor"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)),
 _port(__index(2), __identifier("Inv_log2fact"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("Inv_shiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __
 e0_size(1, 1), __ek_size(1, 1)))

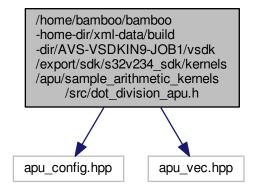
7.42 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample arithmetic kernels/src/dot division apu.h File Reference

Element-wise division of two vectors/matrices. If divisor is zero, a value of zero is returned (in order not to influence following arithmetics).

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
```

Reference 271

Include dependency graph for dot_division_apu.h:



Functions

void dot_division_filter (vec32s *res, vec32s *numerator, vec32s *denominator, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise division of signed 32bit integers.

void computeLog2 (vec08u *log2Fact, vec32s *input, int16s bw, int16s bh, int16s inStrideWidth, int16s out
 StrideWidth)

res[i] = log2(abs(input[i])) + 1, where input is a signed 32bit integer

void computeLog2u (vec08u *log2Fact, vec32u *input, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(input[i]) + 1, where input is a unsigned 32bit integer

• void compute64bitLog2 (vec08u *log2Fact, vec32s *input_high, vec32u *input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(abs(input[i])) + 1, where input is a signed 64bit integer

• void compute64bitLog2u (vec08u *log2Fact, vec32u *input_high, vec32u *input_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

res[i] = log2(input[i]) + 1, where input is a unsigned 64bit integer

• void computeInv_NewtonRaphson (vec32s *invDiv, vec32s *div, vec08u *log2Fact, const int08s shiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise inverse of a signed integer vector using the NewtonRaphson algorithm.

void dot_division_filter_N64s_D32s_Q64s (vec32s *dst_high, vec32u *dst_low, vec32u *dst_rem, vec32s *nom_high, vec32u *nom_low, vec32s *divisor, int16s bw, int16s bh, int16s inStrideWidth, int16s outStride Width)

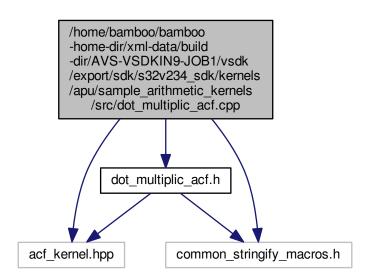
Elementwise division of a 64bit nominator by a 32bit divisor.

7.42.1 Detailed Description

Element-wise division of two vectors/matrices. If divisor is zero, a value of zero is returned (in order not to influence following arithmetics).

7.43 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "common_stringify_macros.h"
#include "dot_multiplic_acf.h"
Include dependency graph for dot multiplic acf.cpp:
```



Functions

- KERNEL_INFO apu_dot_mult_in16s_out32s (" apu_dot_mult_in16s_out32s ", 3, __port(__index(0), _
 __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 __type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek
 __size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_in32s_out32s (" apu_dot_mult_in32s_out32s ", 3, __port(_index(0), _
 _ identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 _ type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek
 _ size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_in32s_out64s (" apu_dot_mult_in32s_out64s ", 4, __port(__index(0), _
 __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 __type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek
 __size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT_High"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __
 identifier("DotMultKn_OUT_Low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_
 data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

Reference 273

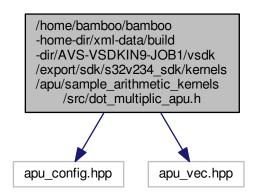
- KERNEL_INFO apu_dot_mult_in32s_in16s_out32s (" apu_dot_mult_in32s_in16s_out32s ", 3, __port(__ index(0), __identifier("DotMultKn_IN_A"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __ e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("DotMultKn_IN_B"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ ek_size(1, 1)), __port(__index(2), __identifier("DotMultKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __ ex_spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_scalar_in08u_out16s (" apu_dot_mult_scalar_in08u_out16s ", 3, __port(_
 __index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __
 e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultScalKn_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL
 __IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_mult_scalar_in32s_out32s (" apu_dot_mult_scalar_in32s_out32s ", 3, __port(_
 __index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __
 e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultScalKn_OUT"),
 __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL
 __IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_left_shift_in16u_out16s (" apu_dot_left_shift_in16u_out16s ", 3, __port(__index(0), __identifier("MultScalKn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data
 _type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultScalKn_OUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("MultScalKn_IN_SCALAR"), __attributes(ACF_ATTR_SCL
 _IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_right_shift_in64s_out64s (" apu_dot_right_shift_in64s_out64s ", 5, __port(_
 index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("RightShift_↔ 64bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0
 __size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("RightShift_64bit_OutHigh"), __attributes(A
 CF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("RightShift_64bit_OutLOW"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __
 identifier("RightShft_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_right_shift_in64s_out32s (" apu_dot_right_shift_in64s_out32s ", 4, __port(_
 index(0), __identifier("RightShift_64bit_InHigh"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("RightShift_04bit_InLOW"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __oe0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("RightShift_32bit_Out"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("RightShift_64bit_ShiftFact"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIometry), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_out32s (" apu_dot_lsh1_in32s_out32s ", 2, __port(__index(0), _
 __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2Kn_OUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), _
 __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_out64s (" apu_dot_lsh1_in32s_out64s ", 3, __port(__index(0), _
 __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_
 type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2Kn_Output_high"),
 __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(2), __identifier("MultBy2Kn_Output_low"), __attributes(ACF_ATTR_VEC_
 OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_dot_lsh1_in32s_Q3_28_out64s (" apu_dot_lsh1_in32s_Q3_28_out64s ", 3, __port(← index(0), __identifier("MultBy2Kn_IN"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), _← e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("MultBy2_Q3_28← Kn_Output_int"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s),

```
 \underline{\hspace{0.5cm}} e0\_size(1,\ 1),\ \underline{\hspace{0.5cm}} ek\_size(1,\ 1)),\ \underline{\hspace{0.5cm}} port(\underline{\hspace{0.5cm}} index(2),\ \underline{\hspace{0.5cm}} identifier("MultBy2\_Q3\_28\_Kn\_Output\_frac"),\ \underline{\hspace{0.5cm}} \Leftrightarrow attributes(ACF\_ATTR\_VEC\_OUT),\ \underline{\hspace{0.5cm}} spatial\_dep(0,\ 0,\ 0,\ 0),\ \underline{\hspace{0.5cm}} e0\_data\_type(d32s),\ \underline{\hspace{0.5cm}} e0\_size(1,\ 1),\ \underline{\hspace{0.5cm}} \Leftrightarrow size(1,\ 1)))
```

7.44 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_multiplic_apu.h File Reference

Element-wise multiplication of two vectors/matrices.

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
Include dependency graph for dot_multiplic_apu.h:
```



Functions

• vbool hasSign (vec32s &a, vec32s &b)

```
sign(a) * sign(b) == -1
```

void change64bitSign (vec32s &highWord, vec32u &lowWord)

In place sign change of a 64bit integer, i.e. a = -a.

void dot_mult_in16s_out32s_filter (vec32s *dst, vec16s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit multiplication => signed 32bit.

void dot_mult_in32s_out32s_filter (vec32s *dst, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 32bit.

void dot_mult_in32s_in16s_out32s_filter (vec32s *dst, vec32s *srcA, vec16s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit and 16bit multiplication => signed 32bit.

• void dot_mult_in32s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *srcA, vec32s *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication => signed 64bit.

• void dot_mult_in32u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA, vec32u *srcB, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Reference

Elementwise unsigned 32bit multiplication => unsigned 64bit.

• void dot_mult_in64s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_

Elementwise signed 64bit multiplication => signed 64bit.

void dot_mult_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_bigh, vec32u *srcB_low, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

low, vec32s *srcB_high, vec32u *srcB_low, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit multiplication => unsigned 64bit.

void dot_mult_scalar_in08u_out16s_filter (vec16s *dst, vec08u *srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 8bit multiplication with a fixed scalar => signed 16bit.

• void dot_mult_scalar_in32s_out32s_filter (vec32s *dst, vec32s *srcA, int32s scalar, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit multiplication with a fixed scalar => signed 32bit.

• void lsh-in16u_out16s_filter (vec16s *upShifted, vec16u *src, vec16s *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 16bit left shift with a signed 16bit shift vector => signed 16bit.

void lsh_in32u_out32u_filter (vec32u *upShifted, vec32u *src, vec08u *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 32bit.

void lsh_in32s_out32s_filter (vec32s *upShifted, vec32s *src, vec08u *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 32bit.

void lsh_in32s_out64s_filter (vec32s *upShifted_high, vec32u *upShifted_low, vec32s *src, vec08u *left←
 ShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit left shift with an unsigned 8bit shift vector => signed 64bit.

void lsh_in32u_out64u_filter (vec32u *upShifted_high, vec32u *upShifted_low, vec32u *src, vec08u *left←
 ShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift with an unsigned 8bit shift vector => unsigned 64bit.

void lsh_in32s_Q3_28_out64s_filter (vec32s *upShifted_int, vec32s *upShifted_frac, vec32s *src, vec08u *leftShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit left shift of a 32bit matrix in Q3_28 format with an unsigned 8bit shift vector => signed 64bit.

• void rsh_in32u_out32u_filter (vec32u *downShift, vec32u *src, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise unsigned 32bit right shift with an unsigned 8bit shift vector => unsigned 32bit.

void rsh_in32s_out32s_filter (vec32s *downShift, vec32s *src, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 32bit right shift with a signed 8bit shift vector => signed 32bit.

void rsh_in64s_out64s_filter (vec32s *dst_high, vec32u *dst_low, vec32s *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

• void rsh_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 64bit.

• void rsh_in64s_out32s_filter (vec32s *dst, vec32s *in_high, vec32u *in_low, vec08u *rightShiftFact, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

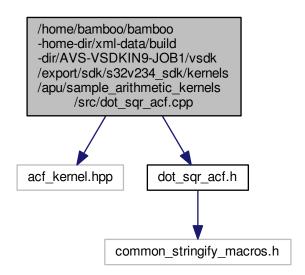
Chunk-wise signed 64bit right shift with a signed 8bit shift vector => signed 32bit.

7.44.1 Detailed Description

Element-wise multiplication of two vectors/matrices.

7.45 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample arithmetic kernels/src/dot sqr acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "dot_sqr_acf.h"
Include dependency graph for dot_sqr_acf.cpp:
```



Functions

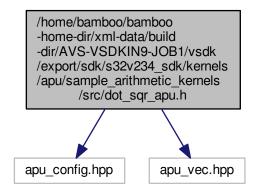
- KERNEL_INFO apu_dot_sqr_in32s_out64u (" apu_dot_sqr_in32s_out64u ", 3, __port(__index(0), __
 identifier("INPUT"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s),
 — e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_High"), __attributes(ACF_AT
 TR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(2), __identifier("OUTPUT_Low"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))
- 7.46 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/dot_sqr_apu.h File Reference

Element-wise square of the input matrix.

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
```

Reference 277

Include dependency graph for dot_sqr_apu.h:



Functions

- vec32u vsqrt 32 (vec32u a)
- void dot_sqr_in16s_out32u_filter (vec32u *dst, vec16s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 16bit square => unsigned 32bit.

void dot_sqr_in32s_out32u_filter (vec32u *dst, vec32s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 32bit.

• void dot_sqr_in32s_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32s *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 32bit square => unsigned 64bit.

• void dot_sqr_in32u_out64u_filter (vec32u *out_high, vec32u *out_low, vec32u *srcA, int16s bw, int16s bh, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 32bit square => unsigned 64bit.

• void dot_sqr_in64s_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32s *srcA_high, vec32u *srcA_← low, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise signed 64bit square => unsigned 64bit.

void dot_sqr_in64u_out64u_filter (vec32u *dst_high, vec32u *dst_low, vec32u *srcA_high, vec32u *srcA_ iow, int16s bw, int16s inStrideWidth, int16s outStrideWidth)

Elementwise unsigned 64bit square => unsigned 64bit.

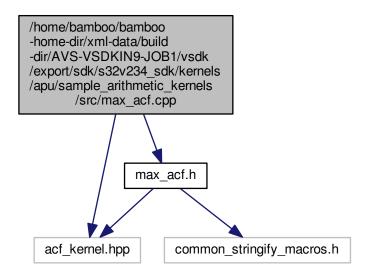
7.46.1 Detailed Description

Element-wise square of the input matrix.

7.47 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/max_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "max_acf.h"
```

Include dependency graph for max_acf.cpp:



Functions

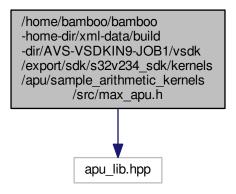
• KERNEL_INFO apu_max (" apu_max ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_
ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __
port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __
e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_0"), __
attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
ek_size(1, 1)))

7.48 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_arithmetic_kernels/src/max_apu.h File Reference

element-wise maximum implementation for APEX

#include "apu_lib.hpp"

Include dependency graph for max_apu.h:



Functions

• void max (vec08u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Element-wise maximum.

7.48.1 Detailed Description

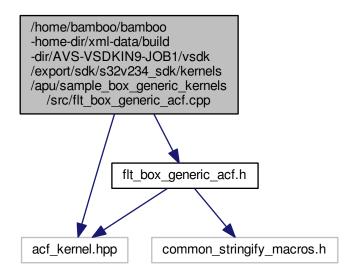
element-wise maximum implementation for APEX

7.49 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.cpp File Reference

ACF metadata and wrapper function for the harris corner.

```
#include "acf_kernel.hpp"
#include "flt_box_generic_acf.h"
```

Include dependency graph for flt_box_generic_acf.cpp:



Functions

- KERNEL_INFO apu_flt_box_generic_16s (" apu_flt_box_generic_16s ", 2, __port(_index(0), __
 identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d16s),
 __e0_size(1, 1), __ek_size(4, 8)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_
 VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 8)))
 ACF metadata for pixel summation.

ACF metadata for pixel summation.

7.49.1 Detailed Description

ACF metadata and wrapper function for the harris corner.

7.49.2 Function Documentation

7.49.2.1 KERNEL_INFO apu_flt_box_generic_16s ("apu_flt_box_generic_16s ", 2, __port(_index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 8)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 8)))

ACF metadata for pixel summation.

See also

UG-10267-03 ACF User Guide, Section 3.2.2

7.49.2.2 KERNEL_INFO harris_box_5x5_16s (" harris_box_5x5_16s " , 2 , __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 8)) , __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(4, 8)))

ACF metadata for pixel summation.

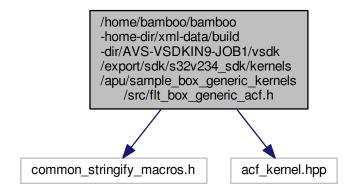
See also

UG-10267-03 ACF User Guide, Section 3.2.2

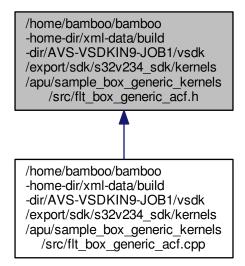
7.50 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_box_generic_kernels/src/flt_box_generic_acf.h File Reference

ACF metadata and wrapper function for the harris corner.

```
#include "common_stringify_macros.h"
#include "acf_kernel.hpp"
Include dependency graph for flt_box_generic_acf.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define FLT_BOX_GENERIC_K apu_flt_box_generic_16s
- #define FLT_BOX_GENERIC_KN XSTR(FLT_BOX_GENERIC_K)
- #define HARRIS_BOX_5X5_K harris_box_5x5_16s
- #define HARRIS_BOX_5X5_KN XSTR(HARRIS_BOX_5X5_K)
- #define FLT_BOX_GENERIC_K_IN "INPUT_0"
- #define FLT_BOX_GENERIC_K_OUT "OUTPUT_0"
- #define HARRIS_BOX_K_IN "INPUT_0"
- #define HARRIS_BOX_K_OUT "OUTPUT_0"

Functions

- void apu_flt_box_generic_16s (kernel_io_desc in0, kernel_io_desc out0)
- void harris_box_5x5_16s (kernel_io_desc in0, kernel_io_desc out0)
- extKernelInfoDecl (apu_flt_box_generic_16s)
- extKernelInfoDecl (harris box 5x5 16s)

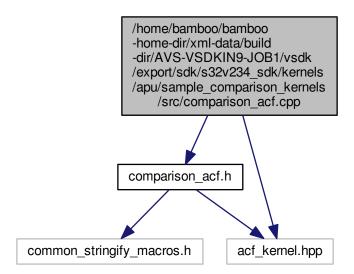
7.50.1 Detailed Description

ACF metadata and wrapper function for the harris corner.

7.51 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/sample_comparison_kernels/src/comparison_acf.cpp File

7.51 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_comparison_kernels/src/comparison_acf.cpp File Reference

```
#include "comparison_acf.h"
#include "acf_kernel.hpp"
Include dependency graph for comparison acf.cpp:
```



Functions

- KERNEL_INFO apu_lower_scalar (" apu_lower_scalar ", 3, __port(__index(0), __identifier("LOWER_KN_I ← N_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_Scalar"), __attributes(ACF_ATTR_SCL_IN ← STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __ ← port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_in16s (" apu_lower_in16s ", 3, __port(__index(0), __identifier("LOWER_KN_ \ IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_ \ IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0 \ _data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lower_in32s (" apu_lower_in32s ", 3, __port(__index(0), __identifier("LOWER_KN_ ← IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATTR_VEC_ ← IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __e0_size(1, 1), __ek_size(1, 1)), __ek_size(1, 1))

 $\underline{\quad} identifier("LOWER_KN_OUT_0"), \\ \underline{\quad} attributes(ACF_ATTR_VEC_OUT), \\ \underline{\quad} spatial_dep(0, 0, 0, 0), \\ \underline{\quad} e0 \longleftrightarrow \\ \underline{\quad} data_type(d08u), \\ \underline{\quad} e0_size(1, 1), \\ \underline{\quad} ek_size(1, 1)))$

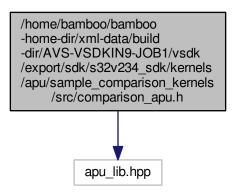
- KERNEL_INFO apu_abs_lower_in32s ("apu_abs_lower_in32s ", 3, __port(__index(0), __identifier("LOWE← R_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_ATT← R_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_lowerEqual_in32s (" apu_lowerEqual_in32s ", 3, __port(__index(0), __identifier("LO ← WER_KN_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __ ← e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LOWER_KN_IN_1"), __attributes(ACF_AT ← TR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ← index(2), __identifier("LOWER_KN_OUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_mask8b (" apu_mask8b ", 3, __port(__index(0), __identifier("MASK_IN_IMG"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(1), __identifier("MASK_IN_MASK"), __attributes(ACF_ATTR_VEC_IN), __
 _ spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), \leftrightarrow
 identifier("MASK_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_\leftrightarrow
 type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_and (" apu_and ", 3, __port(__index(0), __identifier("AND_IN_0"), __attributes(ACF
 _ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0),
 __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("AND_OUT"), __
 _attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)))
- KERNEL_INFO apu_and_in16u_out16u (" apu_and_in16u_out16u ", 3, __port(_index(0), __identifier("A⇔ ND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0⇔ __size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VE⇔ C_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(__⇔ index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0⇔ __data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_and_3Pt_in16u_out16u (" apu_and_3Pt_in16u_out16u ", 4, __port(_index(0), __
 identifier("AND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u),
 __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(\(\to \) index(2), __identifier("AND_IN_2"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __\(\to \) e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("AND_OUT"), __\(\to \) attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __\(\to \) ek_size(1, 1)))
- KERNEL_INFO apu_and_in08u_out16u (" apu_and_in08u_out16u ", 3, __port(_index(0), __identifier("A← ND_IN_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0← __size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("AND_IN_1"), __attributes(ACF_ATTR_VE← C_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_← index(2), __identifier("AND_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0← data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))

N C 1 s id d - - • K	ERNEL_INFO apu_lower_in64s (" apu_lower_in64s ", 5,port(index(0),identifier("LOWER_KN_ID_N_0_HIGH"),attributes(ACF_ATTR_VEC_IN),spatial_dep(0, 0, 0, 0),e0_data_type(d32s),e0_size(1, 1),ek_size(1, 1)),port(index(1),identifier("LOWER_KN_IN_0_LOW"),attributes(ACDE_ATTR_VEC_IN),spatial_dep(0, 0, 0, 0),e0_data_type(d32u),e0_size(1, 1),ek_size(1, 1)),port(index(2),identifier("LOWER_KN_IN_1_HIGH"),attributes(ACF_ATTR_VEC_IN),compatial_dep(0, 0, 0, 0),e0_data_type(d32s),e0_size(1, 1),ek_size(1, 1)),port(index(3),compatial_dep(0, 0, 0, 0),e0_data_type(d32u),e0_size(1, 1),ek_size(1, 1)),port(_index(4),identifier("LOWER_KN_OUT_0"),attributes(ACF_ATTR_VEC_OUT),spatial_dep(0, 0, 0, 0),e0_data_type(d08u),e0_size(1, 1),ek_size(1, 1))) **ERNEL_INFO apu_lower_in64u (" apu_lower_in64u ", 5,port(index(0),identifier("LOWER_KN_CUL_OUT_0"),spatial_dep(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
s id d)),port(index(2),identifier("LOWER_KN_IN_1_HIGH"),attributes(ACF_ATTR_VEC_IN), patial_dep(0, 0, 0, 0),e0_data_type(d32u),e0_size(1, 1),ek_size(1, 1)),port(index(3), dentifier("LOWER_KN_IN_1_LOW"),attributes(ACF_ATTR_VEC_IN),spatial_dep(0, 0, 0, 0),e0_ lata_type(d32u),e0_size(1, 1),ek_size(1, 1)),port(index(4),identifier("LOWER_KN_OUT_0"), attributes(ACF_ATTR_VEC_OUT),spatial_dep(0, 0, 0, 0),e0_data_type(d08u),e0_size(1, 1), ek_size(1, 1)))
7.52	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v2_sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.cpp File Reference
compare functions implementation for APEX	
7.52.1	Detailed Description
compare functions implementation for APEX	
Author A	nca Dima
Version	
Date	
7.53	/home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v2_sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.h File Reference

 ${\tt sdk/kernels/apu/sample_comparison_kernels/src/comparison_apu.cpp\ File}$

element-wise comparison implementation for APEX

#include "apu_lib.hpp"
Include dependency graph for comparison apu.h:



Functions

- void lower (vbool *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

 Elementwise unsigned 8bit "<" operation => bool.
- void lower_scalar (vec08u *dst, vec08u *srcImage, unsigned char scalar, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "<" operation => bool.

void lower_in16s (vbool *dst, vec16s *srcImage0, vec16s *srcImage1, int bw, int bh, int inStrideW, int out
 StrideW)

Elementwise signed 16bit "<" operation => bool.

void lower_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int out
 StrideW)

Elementwise signed 32bit "<" operation => bool.

void absLower_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between the absolute values of the operands => bool.

void absLower_in32s_scalar16u (vbool *dst, vec32s *srcImage0, int16u compVal, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<" operation between an image and a fixed unsigned 16bit scalar => bool.

void lowerEqual_in32s (vbool *dst, vec32s *srcImage0, vec32s *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 32bit "<=" operation => bool.

void lower_in64u (vbool *dst, vec32u *srcImage0_high, vec32u *srcImage0_low, vec32u *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 64bit "<" operation => bool.

• void lower_in64s (vbool *dst, vec32s *srcImage0_high, vec32u *srcImage0_low, vec32s *srcImage1_high, vec32u *srcImage1_low, int bw, int bh, int inStrideW, int outStrideW)

Elementwise signed 64bit "<" operation => bool.

- void mask_kn (vec08u *dst, vec08u *srcImage, vec08u *srcMask, int bw, int bh, int inStrideW, int outStrideW)

 Elementwise unsigned 8bit mask operation => unsigned 8bit.

Elementwise unsigned 8bit "&&" operation => unsigned 8bit.

• void and_in16u_out16u (vec16u *dst, vec16u *srcImage0, vec16u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 16bit "&&" operation => unsigned 16bit.

void and_in08u_out16u (vec16u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit "&&" operation => unsigned 16bit.

Elementwise unsigned 8bit "&&" unsigned 16bit operation => unsigned 16bit.

• void and_3Pt_in16u_out16u (vec16u *dst, vec16u *srcImage0, vec16u *srcImage1, vec16u *srcImage2, int bw, int inStrideW, int outStrideW)

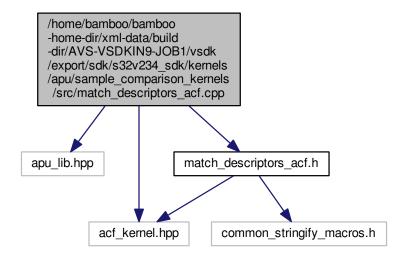
Elementwise unsigned 3-point 16bit "&&" operation => unsigned 16bit.

7.53.1 Detailed Description

element-wise comparison implementation for APEX

7.54 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_acf.cpp File Reference

```
#include <apu_lib.hpp>
#include <acf_kernel.hpp>
#include "match_descriptors_acf.h"
Include dependency graph for match_descriptors_acf.cpp:
```



Functions

• KERNEL_INFO apu_match_descriptors (" apu_match_descriptors ", 5, __port(__index(0), __identifier("I← NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), ← __identifier("INPUT_CONFIG"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), __identifier("OUTPUT_← 0"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(512, 1)), __port(__index(4), __identifier("OUTPUT_1"), __attributes(ACF_ATT← R_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_← size(512, 1)))

7.55 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.cpp File Reference

Binary descriptor matching implementation for APEX.

7.55.1 Detailed Description

Binary descriptor matching implementation for APEX.

Author

Igor Aleksandrowicz

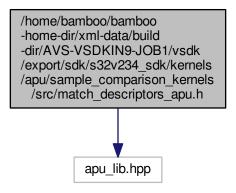
Version

Reference 289

7.56 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_comparison_kernels/src/match_descriptors_apu.h File Reference

binary descriptor matching

```
#include "apu_lib.hpp"
Include dependency graph for match_descriptors_apu.h:
```



Functions

void Match (const vec08u *apcDescriptors0, unsigned int aDescriptor0Count, const vec08u *apc
 Descriptors1, unsigned int aDescriptor1Count, int16s *apMatches0, int16s *apMatches1, int08u aThreshold, int08u aRangeCheck)

Matches binary descriptors.

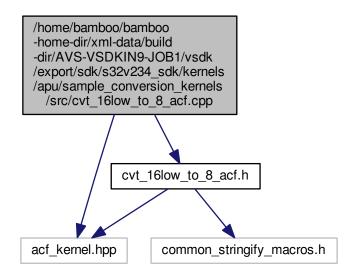
7.56.1 Detailed Description

binary descriptor matching

7.57 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "cvt_16low_to_8_acf.h"
```

Include dependency graph for cvt_16low_to_8_acf.cpp:



Functions

- KERNEL_INFO apu_16low_to_8 (" apu_16low_to_8 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("APU_16LOWTO8_OUT"), __attributes(ACF_ATTR_VEC_OUT),
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- 7.58 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.cpp File Reference

16low_to_8 implementation for APEX

7.58.1 Detailed Description

16low_to_8 implementation for APEX

Author

Igor Aleksandrowicz

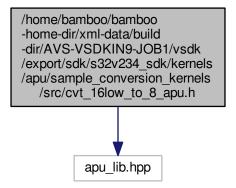
Version

Date

7.59 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.h File

Reference
7.59 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234
_sdk/kernels/apu/sample_conversion_kernels/src/cvt_16low_to_8_apu.h File Reference

```
#include "apu_lib.hpp"
Include dependency graph for cvt_16low_to_8_apu.h:
```



Functions

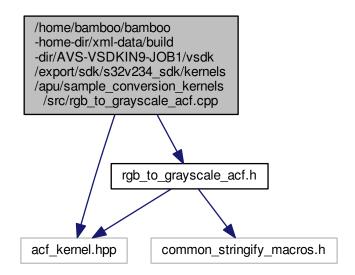
• void f16low_to_8 (vec08u *dst, vec16u *src, int bw, int bh)

Extracts the lower bytes.

7.60 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "rgb_to_grayscale_acf.h"
```

Include dependency graph for rgb_to_grayscale_acf.cpp:



Functions

- KERNEL_INFO apu_rgb_to_grayscale (" apu_rgb_to_grayscale ", 2, __port(__index(0), __identifier("INP \cup UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), \cup __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- 7.61 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.cpp File Reference

rgb_to_grayscale implementation for APEX

7.61.1 Detailed Description

rgb_to_grayscale implementation for APEX

Author

Igor Aleksandrowicz

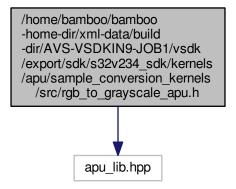
Version

Date

7.62 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.h File

Reference
7.62 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234
_sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_grayscale_apu.h File Reference

```
#include "apu_lib.hpp"
Include dependency graph for rgb to grayscale apu.h:
```



Functions

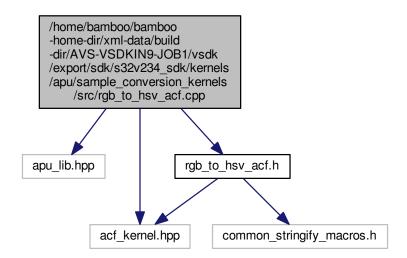
void rgb_to_grayscale (vec08u *apDest, const vec08u *apcSrc, int aBlockWidth, int aBlockHeight, int a
 —
 OutputSpan, int aInputSpan)

Transforms RGB to grayscale.

7.63 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_acf.cpp File Reference

```
#include "apu_lib.hpp"
#include "acf_kernel.hpp"
#include "rgb_to_hsv_acf.h"
```

Include dependency graph for rgb_to_hsv_acf.cpp:



Functions

- KERNEL_INFO apu_rgb_to_hsv_hue_sat (" apu_rgb_to_hsv_hue_sat ", 3, __port(_index(0), __
 identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(3, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUT_SAT"), __attributes(ACF_ATT
 R_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
 __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_rgb_to_hsv_hue_sat_grey ("apu_rgb_to_hsv_hue_sat_grey ", 4, __port(_index(0), _
 __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u),
 __e0_size(3, 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_
 VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(
 __index(2), __identifier("OUT_HUE"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e
 e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("OUT_GREY"), __e
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __e
 ek_size(1, 1)))
- KERNEL_INFO apu_rgb_to_hsv_svr (" apu_rgb_to_hsv_svr ", 4, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(3, 1), __ek __size(1, 1)), __port(__index(1), __identifier("OUT_SAT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT_VAL"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_data_type(d08u), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

 UT, __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.64 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_
sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.cpp File

Reference
7.64 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234
_sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.cpp File Reference

rgb_to_grayscale implementation for APEX

7.64.1 Detailed Description

rgb_to_grayscale implementation for APEX

Author

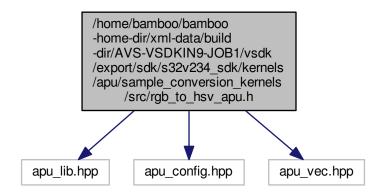
Igor Aleksandrowicz

Version

Date

7.65 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_conversion_kernels/src/rgb_to_hsv_apu.h File Reference

```
#include "apu_lib.hpp"
#include "apu_config.hpp"
#include "apu_vec.hpp"
Include dependency graph for rgb_to_hsv_apu.h:
```



Functions

void rgb_to_hsv_sat (vec08u *apSat, const vec08u *apcSrc, int aBlockWidth, int aBlockHeight, int aOutput
 — Span, int aInputSpan)

Transforms RGB to HSV => S.

void rgb_to_hsv_hue_sat (vec16u *apHue, vec08u *apSat, const vec08u *apcSrc, int aBlockWidth, int a
 —
 BlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to HSV => (Hue,Sat)

• void rgb_to_hsv_hue_sat_grey (vec16u *apHue, vec08u *apSat, vec08u *grey, const vec08u *apcSrc, int aBlockWidth, int aBlockHeight, int aOutputSpan, int aInputSpan)

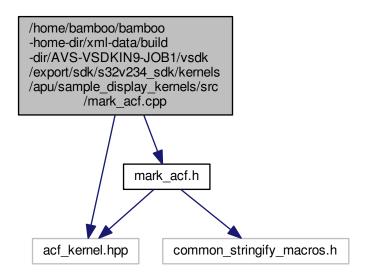
Transforms RGB to HSV, Grey => (Hue, Sat, Grey)

void rgb_to_hsv_svr (vec08u *apSat, vec08u *apVal, vec08u *apRed, const vec08u *apcSrc, int aBlock
Width, int aBlockHeight, int aOutputSpan, int aInputSpan)

Transforms RGB to HSV => (S, V, Red)

7.66 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample display kernels/src/mark acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "mark_acf.h"
Include dependency graph for mark acf.cpp:
```



Functions

7.67 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/sample_display_kernels/src/mark_apu.cpp File

Reference 7.67 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_apu.cpp File Reference

mark implementation for APEX

7.67.1 Detailed Description

mark implementation for APEX

Author

Igor Aleksandrowicz

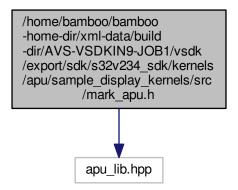
Version

Date

7.68 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_apu.h File Reference

image marking implementation for APEX

#include "apu_lib.hpp"
Include dependency graph for mark_apu.h:



Functions

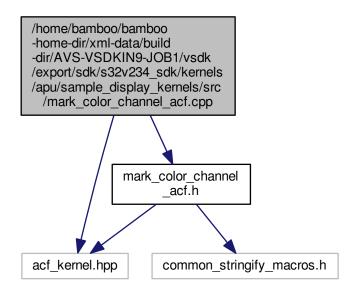
void mark (vec08u *dst, vec08u *srcImage, vec08u *srcMarker, int bw, int bh, int sstride, int mstride)
 Marks the image.

7.68.1 Detailed Description

image marking implementation for APEX

7.69 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "mark_color_channel_acf.h"
Include dependency graph for mark_color_channel_acf.cpp:
```



Functions

7.70 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File

7.70 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.cpp File Reference

mark_color_channel implementation for APEX

7.70.1 Detailed Description

mark_color_channel implementation for APEX

Author

Igor Aleksandrowicz

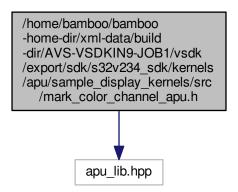
Version

Date

7.71 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_display_kernels/src/mark_color_channel_apu.h File Reference

color channel image marking implementation for APEX

#include "apu_lib.hpp"
Include dependency graph for mark_color_channel_apu.h:



Functions

• void mark_color_channel (vec08u *dst, vec08u *srcImage, vec08u *srcMarker, int bw, int bh, int08u channel, int inStride, int inMarkerStride, int outStride)

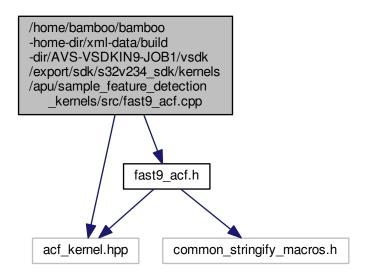
Marks a color channel of the image.

7.71.1 Detailed Description

color channel image marking implementation for APEX

7.72 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_acf.cpp File Reference

```
#include <acf_kernel.hpp>
#include "fast9_acf.h"
Include dependency graph for fast9_acf.cpp:
```



Functions

• KERNEL_INFO apu_fast9 (" apu_fast9 ", 3, __port(__index(0), __identifier("IN_Img"), __attributes(ACF_ ← ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), _ ← _ port(__index(1), __identifier("OUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("IN_Thr"), __ ← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(1, 1)))

7.73 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_← sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_apu.cpp File

7.73 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_apu.cpp File Reference

FAST 9 corner detection implementation for APEX.

7.73.1 Detailed Description

FAST 9 corner detection implementation for APEX.

Author

Igor Aleksandrowicz

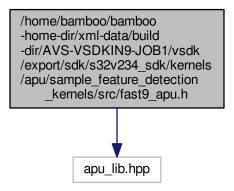
Version

Date

7.74 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_feature_detection_kernels/src/fast9_apu.h File Reference

FAST 9 corner detection implementation for APEX.

#include "apu_lib.hpp"
Include dependency graph for fast9_apu.h:



Functions

void apu_fast9_unsuppressed_score (const vec08u *apcSrc, vec08u *apDst, int aSourceStride, int a⇔ DestinationStride, int aBlockWidth, int aBlockHeight, int08u aThreshold)

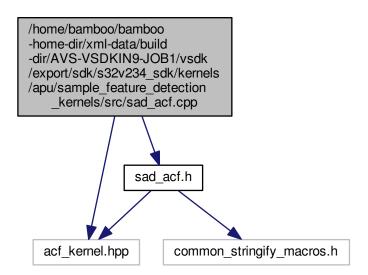
FAST9 corner detection.

7.74.1 Detailed Description

FAST 9 corner detection implementation for APEX.

7.75 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_feature_detection_kernels/src/sad_acf.cpp File Reference

```
#include <acf_kernel.hpp>
#include "sad_acf.h"
Include dependency graph for sad_acf.cpp:
```

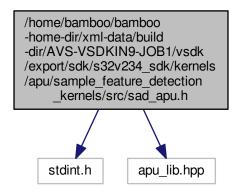


Functions

- KERNEL_INFO apu_sad (" apu_sad ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_A← TTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(4, 4)), __port(__index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(8, 8)), __port(__index(2), __identifier("OUTPUT_← 0"), __attributes(ACF_ATTR_VEC_OUT_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_← size(4, 1), __ek_size(1, 1)))
- 7.76 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/sample_feature_detection_kernels/src/sad_apu.h File Reference
- SAD Sum of absolute Differences implementation for APEX.

```
#include <stdint.h>
#include "apu_lib.hpp"
```

Include dependency graph for sad_apu.h:



Functions

void apu_sad_impl (vec08u *lpvln0, int16_t lStrideln0, int16_t lChunkWidthln0, int16_t lChunkHeight
 In0, vec08u *lpvln1, int16_t lStrideln1, int16_t lChunkWidthln1, int16_t lChunkHeightln1, vec32u *lpvOut0,
 int16_t lStrideOut0, int16_t lChunkWidthOut0, int16_t lChunkHeightOut0)

Sum of absolute differences. Store the minimum of all differences and the location of the minimum in image0.

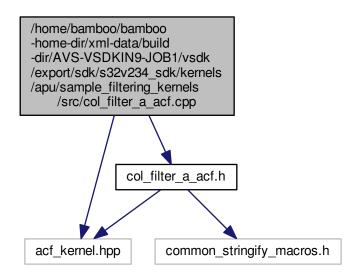
7.76.1 Detailed Description

SAD - Sum of absolute Differences implementation for APEX.

7.77 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "col_filter_a_acf.h"
```

Include dependency graph for col_filter_a_acf.cpp:



Functions

KERNEL_INFO col_filter (" col_filter ", 3, __port(__index(0), __identifier("SRC"), __attributes(ACF_ATTR ← _ VEC_IN), __spatial_dep(COL_PADDING, COL_PADDING, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_← FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_COLS, 1)), __port(__index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __← e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Variables

- const int FILTER COLS = 3
- const int COL_PADDING = FILTER_COLS >> 1

7.78 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.cpp File Reference

Filtering with general filter image columns - implementation for APEX.

7.78.1 Detailed Description

Filtering with general filter image columns - implementation for APEX.

Author

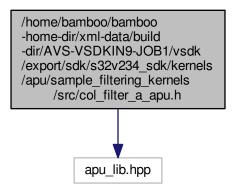
CGV

Date

7.79 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample_filtering_kernels/src/col_filter_a_apu.h File Reference

A column filter implementation for the APU.

#include "apu_lib.hpp"
Include dependency graph for col_filter_a_apu.h:



Functions

• void col_filter (vec08u *dst, int dstStride, const vec08u *src, int srcStride, int rows, int cols, const unsigned char *filter, int filterSize, int filterQ)

Apply a column filter to an image.

7.79.1 Detailed Description

A column filter implementation for the APU.

Filter an image column with a certain filter.

7.79.2 The Column Filter

A column filter is a 1-dimension filter applied to an image where each pixel becomes a weighted sum of itself and neighbouring pixels in the same row. The weighted sum is determined by a set of filter coefficients. The filter are pixel-centered (i.e. the middle coefficient lines up with the pixel under consideration), so the number of coefficients must be odd. For example, consider the column filter with $N_c = 5$ shown below

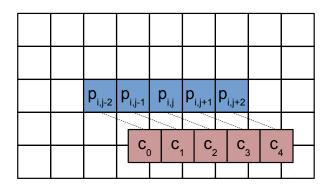


Figure 7.1: A 5-coefficient column filter centered on pixel p_ij

The filter, shown in red, has five columns and a single row of coefficients. If we apply this filter to pixel p_{ij} , the first filter coefficient c_0 lines up with $p_{i,j-2}$, the second coefficient c_1 lines up with $p_{i,j-1}$, and so on. So the weighted sum is

$$p'_{ij} \equiv c_0 p_{i,j-2} + c_1 p_{i,j-1} + c_2 p_{i,j} + c_3 p_{i,j+1} + c_4 p_{i,j+2}$$

In order to filter pixels at the left and right edges of the image, we must pad the image. The number of columns padded on either side of the image is

$$N_{col\ padding} = floor(N_c/2)$$

and so the total number of columns in the padded image is $N_{cols} + 2N_{col\ padding}$, where N_{cols} is the original number of image columns. Note that each row of the image is filtered independently of the others and the no row padding is required.

The filter coefficients are expressed as 8-bit unsigned fixed point numbers with Q fractional bits, where Q is chosen by the user.

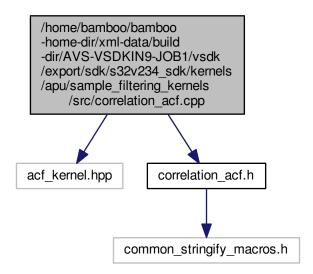
See also

pagFixedPoint

7.80 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample filtering kernels/src/correlation acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "correlation_acf.h"
```

Include dependency graph for correlation_acf.cpp:



Functions

- KERNEL_INFO apu_scharr_x (" apu_scharr_x ", 2, __port(_index(0), __identifier("INPUT_Img"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_scharr_y (" apu_scharr_y ", 2, __port(__index(0), __identifier("INPUT_Img"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16s), __e0_size(1, 1), __
 ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), __
 spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation (" apu_correlation ", 5, __port(__index(0), __identifier("INPUT_Img"), \(\) attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_VEC_OUT), \(\) __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(__index(3), __identifier("INPU\(\) _T_FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_\(\) _data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __\(\) _e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_1x3 (" apu_correlation_1x3 ", 5, __port(__index(0), __identifier("INPU← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V←

EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ \hookleftarrow index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 3)), __port(__index(3), __identifier("INPUT \hookleftarrow _FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data \hookleftarrow _type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFl"), __ \hookleftarrow attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_ \hookleftarrow size(1, 1), __ek_size(1, 1)))

- KERNEL_INFO apu_correlation_3x1 (" apu_correlation_3x1 ", 5, __port(__index(0), __identifier("INPU \cup T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 0, 0), __e0_data_type(d08u), __e0_\cup size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V \cup EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__\cup index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 1)), __port(__index(3), __identifier("INPUT \cup TilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data \cup type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFI"), __\cup attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_\cup size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_3x3 (" apu_correlation_3x3 ", 5, __port(__index(0), __identifier("INPU← T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__← index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(3, 3)), __port(__index(3), __identifier("INPUT← __tilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data← __type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_← size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_5x5 (" apu_correlation_5x5 ", 5, __port(__index(0), __identifier("INPU \cup T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_\cup size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V \cup EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__\cup index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(5, 5)), __port(__index(3), __identifier("INPUT \cup FilterScale"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data \cup type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("INPUT_FiltSymmFI"), __\cup attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_\cup size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_correlation_7x7 (" apu_correlation_7x7 ", 5, __port(__index(0), __identifier("INPU \cup T_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(3, 3, 3, 3), __e0_data_type(d08u), __e0_\cup size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT_Img"), __attributes(ACF_ATTR_V \cup EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__\cup index(2), __identifier("Corr_IN_Filter"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(7, 7)), __port(__index(3), __identifier("INPUT \cup Index(1), __spatial_dep(0, 0, 0, 0), __e0_data \cup type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(4), __identifier("INPUT_FiltSymmFI"), __\cup attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_\cup size(1, 1), __ek_size(1, 1)))

7.81 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 sdk/kernels/apu/sample filtering kernels/src/correlation apu.cpp File Reference

Convolution with general filter 1D/2D.

7.81.1 Detailed Description

Convolution with general filter 1D/2D.

Author

Anca Dima

Version

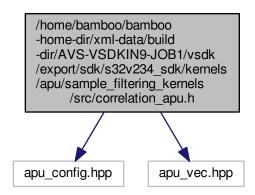
Date

7.82 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/correlation_apu.h File Reference

Correlation with general filter 1D/2D.

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
```

Include dependency graph for correlation_apu.h:



Macros

- #define inputFiltUpScale 3
- #define scharrFiltUpscale 1

Typedefs

• typedef void(* **corrKernelPtr**)(vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Functions

· void initFilters ()

Initializes the array of filter function pointers.

void performCorrelation (int16u filterFlags, vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s *filter← Coefs)

Computes time optimized the correlation with a general filter.

- void correlation_filter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)
- void correlation__antisymXfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric X filter.

void correlation_symXfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric X filter.

• void correlation_symYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric Y filter.

void correlation__antisymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric Y filter.

void correlation_symXYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with a symmetric in X and Y filter.

void correlation_symXantisymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter← Coefs)

Computes the correlation with a symmetric in X and anti-symmetric in Y filter.

void correlation_symXantisymYfilter_16s (vec16s *dst, vec16s *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s scaleFact, const int16s *filter← Coefs)

Computes the 16bit correlation with a symmetric in X and anti-symmetric in Y filter.

void correlation__antisymXsymYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter

Coefs)

Computes the correlation with an anti-symmetric in X and symmetric in Y filter.

void correlation__antisymXsymYfilter_16s (vec16s *dst, vec16s *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filter← Coefs)

Computes the 16bit correlation with an anti-symmetric in X and symmetric in Y filter.

• void correlation__antisymXYfilter (vec16s *dst, vec08u *src, int16s sstr, int16s bw, int16s bh, int16s destBw, int16s xSkip, int16s ySkip, int16u filtWidth, int16u filtHeight, int16s filtScaling, const int16s *filterCoefs)

Computes the correlation with an anti-symmetric in X and anti-symmetric in Y filter.

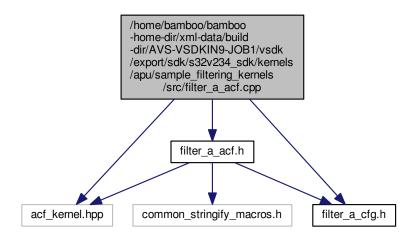
7.82.1 Detailed Description

Correlation with general filter 1D/2D.

7.83 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234_ sdk/kernels/apu/sample_filtering_kernels/src/filter_a_acf.cpp File

Reference
7.83 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234
_sdk/kernels/apu/sample_filtering_kernels/src/filter_a_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "filter_a_acf.h"
#include "filter_a_cfg.h"
Include dependency graph for filter a acf.cpp:
```



Functions

KERNEL_INFO apu_filter_a ("apu_filter_a ", 3, __port(__index(0), __identifier("INPUT_0"), __attributes(A← CF_ATTR_VEC_IN), __spatial_dep(3 >> 1, 3 >> 1, 3 >> 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_COEF"), __attributes(ACF_ATTR_SCL_IN_S← TATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(3 *3, 1)), _← port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.84 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.cpp File Reference

Filtering with general filter 1D/2D implementation for APEX.

7.84.1 Detailed Description

Filtering with general filter 1D/2D implementation for APEX.

Author

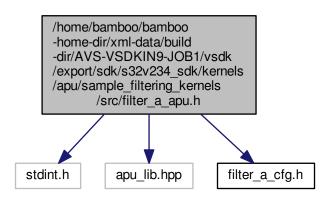
Version

Date

7.85 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filter_a_apu.h File Reference

filtering with general filter 1D/2D

```
#include <stdint.h>
#include "apu_lib.hpp"
#include "filter_a_cfg.h"
Include dependency graph for filter_a_apu.h:
```



Functions

void apu_filter_a_impl (vec08u *src, int16_t sstr, uint8_t *coef, vec08u *dst, int16_t dstr, int16_t bw, int16_t bh)

General filtering function with a 1D/2D-filter.

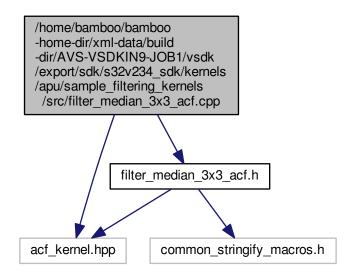
7.85.1 Detailed Description

filtering with general filter 1D/2D

7.86 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "filter_median_3x3_acf.h"
```

Include dependency graph for filter_median_3x3_acf.cpp:



Functions

KERNEL_INFO median_3x3_8bpp (" median_3x3_8bpp ", 2, __port(_index(0), __identifier("DST"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)), __port(_index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Filtering with a Median 3x3-filter.

7.87 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.cpp File Reference

3x3 Median filter implementation for APEX

7.87.1 Detailed Description

3x3 Median filter implementation for APEX

Author

Igor Aleksandrowicz

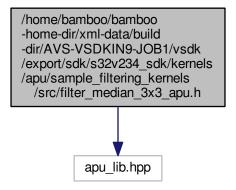
Version

Date

7.88 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filter_median_3x3_apu.h File Reference

3x3 Median filter implementation for APEX

```
#include <apu_lib.hpp>
Include dependency graph for filter_median_3x3_apu.h:
```



Functions

• void apu_flt_median_3x3 (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

Filter with a 3x3 median filter.

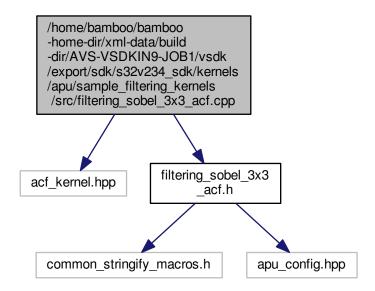
7.88.1 Detailed Description

3x3 Median filter implementation for APEX

7.89 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "filtering_sobel_3x3_acf.h"
```

Include dependency graph for filtering_sobel_3x3_acf.cpp:



Functions

KERNEL_INFO sobel_3x3_8bpp (" sobel_3x3_8bpp ", 2, __port(_index(0), __identifier("SRC"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

7.90 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.cpp File Reference

3x3 Sobel filter implementation for APEX

7.90.1 Detailed Description

3x3 Sobel filter implementation for APEX

Author

Anca Dima

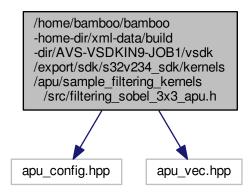
Version

Date

7.91 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/filtering_sobel_3x3_apu.h File Reference

3x3 Sobel filter implementation for APEX

```
#include "apu_config.hpp"
#include "apu_vec.hpp"
Include dependency graph for filtering_sobel_3x3_apu.h:
```



Functions

- void apu_flt_sobel_3x3_x (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

 Calculate first order derivative x using sobel 3x3.
- void apu_flt_sobel_3x3_y (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

 Calculate first order derivative y using sobel 3x3.
- void apu_flt_sobel_3x3 (vec08u *dst, int dstr, const vec08u *src, int sstr, int bw, int bh)

 Calculate sum of absolute values of first order derivatives x and y using sobel 3x3.

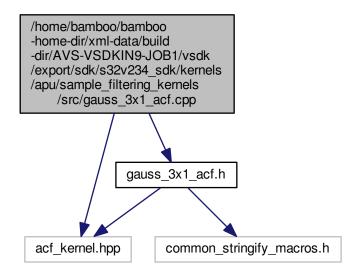
7.91.1 Detailed Description

3x3 Sobel filter implementation for APEX

7.92 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "gauss_3x1_acf.h"
```

Include dependency graph for gauss_3x1_acf.cpp:



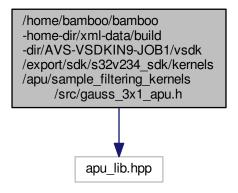
Functions

KERNEL_INFO apu_gauss_3x1 (" apu_gauss_3x1 ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.93 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x1_apu.h File Reference

element-wise addition implementation for APEX

```
#include "apu_lib.hpp"
Include dependency graph for gauss_3x1_apu.h:
```



Functions

• void gauss_3x1 (vec08u *dst, vec08u *srcImage0, int bw, int bh, int inStrideW, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

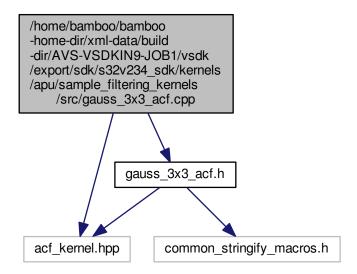
7.93.1 Detailed Description

element-wise addition implementation for APEX

7.94 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_acf.cpp File Reference

```
#include <acf_kernel.hpp>
#include "gauss_3x3_acf.h"
```

Include dependency graph for gauss_3x3_acf.cpp:



Functions

KERNEL_INFO apu_gauss_3x3 (" apu_gauss_3x3 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.95 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_apu.cpp File Reference

3x3 Gauss filter

7.95.1 Detailed Description

3x3 Gauss filter

Author

Igor Aleksandrowicz

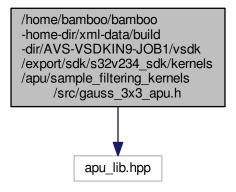
Version

Date

7.96 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_3x3_apu.h File Reference

3x3 Gaussian filter implementation for APEX

```
#include <apu_lib.hpp>
Include dependency graph for gauss 3x3 apu.h:
```



Functions

 void apu_gauss_3x3 (vec08u *apOut, const vec08u *apcIn, int aOutStride, int aInStride, int aTileWidth, int aTileHeight)

3x3 gaussian filter.

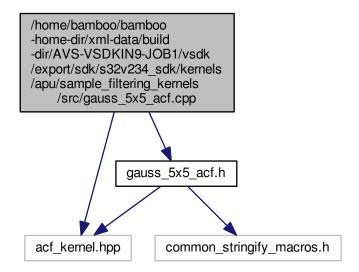
7.96.1 Detailed Description

3x3 Gaussian filter implementation for APEX

7.97 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "gauss_5x5_acf.h"
```

Include dependency graph for gauss_5x5_acf.cpp:



Functions

KERNEL_INFO apu_gauss_5x5 (" apu_gauss_5x5 ", 2, __port(_index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(2, 2, 2, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.98 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.cpp File Reference

5x5 Gauss filter implementation for APEX

7.98.1 Detailed Description

5x5 Gauss filter implementation for APEX

Author

Igor Aleksandrowicz

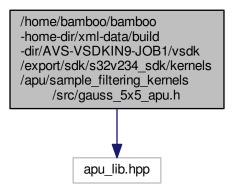
Version

Date

7.99 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234 _sdk/kernels/apu/sample_filtering_kernels/src/gauss_5x5_apu.h File Reference

5x5 Gaussian filter implementation for APEX

```
#include "apu_lib.hpp"
Include dependency graph for gauss_5x5_apu.h:
```



Functions

• void Gauss_5x5__filter (vec08u *dst, vec08u *src, int sstr, int bw, int bh)

5x5 gaussian filter.

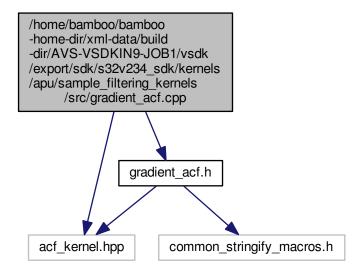
7.99.1 Detailed Description

5x5 Gaussian filter implementation for APEX

7.100 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/gradient_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "gradient_acf.h"
```

Include dependency graph for gradient acf.cpp:



Functions

- KERNEL_INFO apu_gradient_out08s (" apu_gradient_out08s ", 3, __port(__index(0), __identifier("IN← PUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_← size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUT_GX"), __attributes(ACF_ATTR_VEC_← OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)), __port(_ ← index(2), __identifier("OUT_GY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_← data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_gradient_abs (" apu_gradient_abs ", 4, __port(__index(0), __identifier("INPUT_0"), _
 _ attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek
 _ size(1, 1)), __port(__index(1), __identifier("OUT_GX"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUT_GY"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08s), __e0_
 size(1, 1), __ek_size(1, 1)), __port(__index(3), __identifier("OUT_ABSSUM"), __attributes(ACF_ATTR_V
 EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_gr_abs ("apu_gr_abs ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("OUT_ABSSUM"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.101 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.cpp File Reference

gradient implementation for APEX

7.101.1 Detailed Description

gradient implementation for APEX

Author

Igor Aleksandrowicz

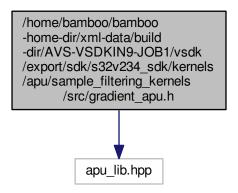
Version

Date

7.102 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/gradient_apu.h File Reference

image gradient implementation for APEX

#include "apu_lib.hpp"
Include dependency graph for gradient_apu.h:



Functions

void apuGradient (vec16s *apcSobelX, vec16s *apcSobelY, const vec08u *apInput, int aBlockWidth, int a
 —
 BlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

void apuGradient_out08s (vec08s *apcSobelX, vec08s *apcSobelY, const vec08u *apInput, int aBlockWidth, int aBlockHeight, int aStride)

Image gradient. ==> GradX, GradY.

void apuGradientAbs (vec08s *apcSobelX, vec08s *apcSobelY, vec08u *apcAbsSum, const vec08u *apcAbsSum, cons

Image gradient ==> GradX, GradY, |GradX|+|GradY|.

Image gradient absulute sum ==> |GradX| + |GradY|.

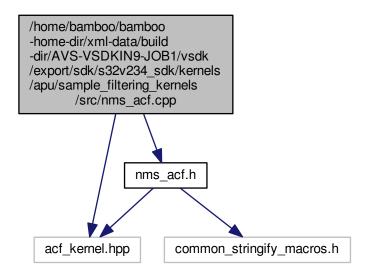
7.102.1 Detailed Description

image gradient implementation for APEX

7.103 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 sdk/kernels/apu/sample filtering kernels/src/nms acf.cpp File Reference

```
#include <acf_kernel.hpp>
#include "nms_acf.h"
```

Include dependency graph for nms_acf.cpp:



Functions

- KERNEL_INFO apu_nms (" apu_nms ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(ACF_
 ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __
 port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_nms16 (" apu_nms16 ", 2, __port(__index(0), __identifier("INPUT_0"), __attributes(A← CF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)),

```
__port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
```

7.104 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.cpp File Reference

Non-maximum supression filter implementation for APEX.

7.104.1 Detailed Description

Non-maximum supression filter implementation for APEX.

Author

Igor Aleksandrowicz

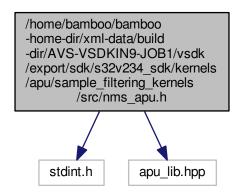
Version

Date

7.105 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/nms_apu.h File Reference

non-maximum suppression implementation for APEX

```
#include <stdint.h>
#include <apu_lib.hpp>
Include dependency graph for nms_apu.h:
```



Reference 327 Functions

 void apu_nms_impl (const vec08u *apcln, vec08u *apOut, int alnStride, int aOutStride, int aTileWidth, int aTileHeight)

Non-maximum suppression.

void apu_nms16 (const vec16u *apcln, vec16u *apOut, int alnStride, int aOutStride, int aTileWidth, int a
 —
 TileHeight)

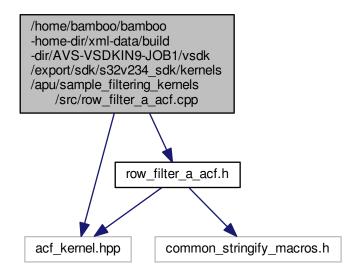
Non-maximum suppression, 16-bit.

7.105.1 Detailed Description

non-maximum suppression implementation for APEX

7.106 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "row_filter_a_acf.h"
Include dependency graph for row_filter_a_acf.cpp:
```



Functions

KERNEL_INFO row_filter (" row_filter ", 3, __port(_index(0), __identifier("SRC"), __attributes(ACF_ATT← R_VEC_IN), __spatial_dep(0, 0, ROW_PADDING, ROW_PADDING), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(1), __identifier("COEFFS"), __attributes(ACF_ATTR_SCL_IN_STAT← IC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(FILTER_ROWS, 1)), __port(_index(2), __identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

Variables

- const int **FILTER_ROWS** = 5
- const int ROW_PADDING = FILTER_ROWS >> 1

7.107 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_apu.cpp File Reference

Filtering with general filter image rows - implementation for APEX.

7.107.1 Detailed Description

Filtering with general filter image rows - implementation for APEX.

Author

CGV

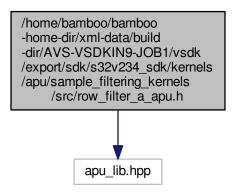
Version

Date

7.108 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/row_filter_a_apu.h File Reference

Row filter implementation for APU2.

```
#include "apu_lib.hpp"
Include dependency graph for row_filter_a_apu.h:
```



• void row_filter_impl (vec08u *dst, int dstStride, const vec08u *src, int srcStride, int rows, int cols, const unsigned char *filter, int filterSize, int filterQ)

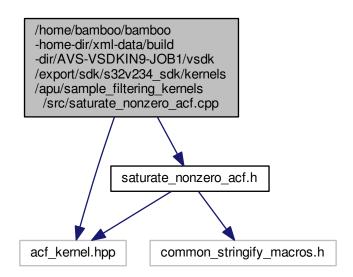
7.108.1 Detailed Description

Functions

Row filter implementation for APU2.

7.109 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "saturate_nonzero_acf.h"
Include dependency graph for saturate_nonzero_acf.cpp:
```



Functions

- KERNEL_INFO apu_saturate_nonzero (" apu_saturate_nonzero ", 2, __port(_index(0), __identifier("INP
 UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1,
 1), __ek_size(1, 1)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT),
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_binarize (" apu_binarize ", 2, __port(__index(0), __identifier("BIN_IN"), __attributes(A
 CF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)),
 __port(__index(1), __identifier("BIN_OUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0),
 __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.110 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.cpp File Reference

saturate nonzero implementation for APEX

7.110.1 Detailed Description

saturate_nonzero implementation for APEX

Author

Igor Aleksandrowicz

Version

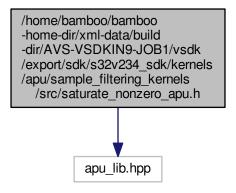
Date

7.111 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_filtering_kernels/src/saturate_nonzero_apu.h File Reference

non-zero saturation implementation for APEX

#include "apu_lib.hpp"

Include dependency graph for saturate_nonzero_apu.h:



Functions

• void saturate_nonzero (vec08u *dst, vec08u *src, int dstr, int sstr, int bw, int bh)

Non-zero pixel saturation.

void binarize (vec08u *dst, vec32u *src, int, int, int bw, int bh)

Non-zero pixel binarization.

• void mask (vec16s *dstX, vec16s *dstY, vec32u *srcFlags, vec16s *inX, vec16s *inY, int, int, int bw, int bh)

Masking to zero with the srcFlags.

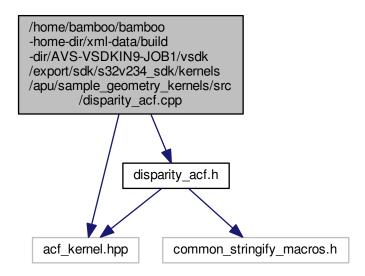
7.111.1 Detailed Description

non-zero saturation implementation for APEX

7.112 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/disparity_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "disparity_acf.h"
```

Include dependency graph for disparity_acf.cpp:



Functions

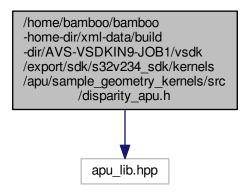
```
• KERNEL_INFO apu_disparity (" apu_disparity ", 3, __port(_index(0), __identifier("INPUT_0"), __ 
attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_
size(1, 1)), __port(_index(1), __identifier("INPUT_1"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 64, 1, 2), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __identifier("OUT 
PUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_
size(1, 1), __ek_size(1, 1)))
```

7.113 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/disparity_apu.h File Reference

element-wise addition implementation for APEX

#include "apu_lib.hpp"

Include dependency graph for disparity_apu.h:



Functions

 void disparity (vec08u *dst, vec08u *srcImage0, vec08u *srcImage1, int bw, int bh, int cw, int ch, int in← StrideW0, int inStrideW1, int outStrideW)

Elementwise unsigned 8bit addition => unsigned 16bit.

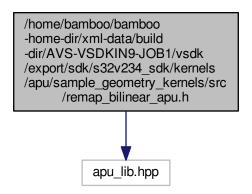
7.113.1 Detailed Description

element-wise addition implementation for APEX

7.114 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/remap_bilinear_apu.h File Reference

element-wise interpolation between pixels of an image for APEX

```
#include "apu_lib.hpp"
Include dependency graph for remap_bilinear_apu.h:
```



Functions

• void remap_bilinear_rgb_impl (vec32u *dst, vec32u *src, vec16u *offset, vec08u *delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

• void remap_bilinear_grayscale_impl (vec08u *dst, vec08u *src, vec16u *offset, vec08u *delta, int sstride, int dstride, int bw, int bh)

Elementwise bilinear interpolation.

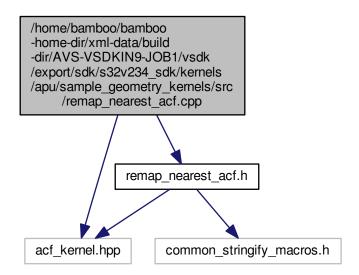
7.114.1 Detailed Description

element-wise interpolation between pixels of an image for APEX

7.115 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "remap_nearest_acf.h"
```

Include dependency graph for remap_nearest_acf.cpp:



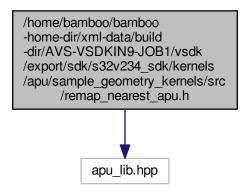
Functions

• KERNEL_INFO remap_nearest_grayscale (" remap_nearest_grayscale ", 3, __port(__index(0), __ identifier("DST"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("SRC"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __ identifier("OFFSET"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))

7.116 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/remap_nearest_apu.h File Reference

element-wise interpolation between pixels of an image for APEX

```
#include "apu_lib.hpp"
Include dependency graph for remap_nearest_apu.h:
```



Functions

void remap_nearest_grayscale_impl (vec08u *dst, vec08u *src, vec16u *offset, int sstride, int dstride, int bw, int bh)

Elementwise nearest neighbor interpolation.

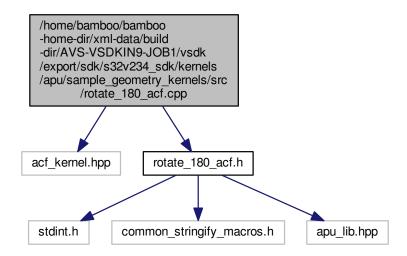
7.116.1 Detailed Description

element-wise interpolation between pixels of an image for APEX

7.117 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "rotate_180_acf.h"
```

Include dependency graph for rotate_180_acf.cpp:



Functions

KERNEL_INFO apu_rotate_180 (" apu_rotate_180 ", 2, __port(__index(0), __identifier("INPUT"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek
 _size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.118 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.cpp File Reference

Rotate an image by 180 deg implementation for APEX.

7.118.1 Detailed Description

Rotate an image by 180 deg implementation for APEX.

Author

Igor Aleksandrowicz

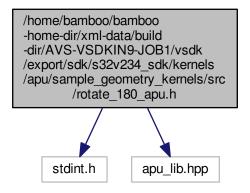
Version

Date

7.119 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_geometry_kernels/src/rotate_180_apu.h File Reference

180-degree rotation implementation for APEX

```
#include <stdint.h>
#include "apu_lib.hpp"
Include dependency graph for rotate_180_apu.h:
```



Functions

• void rotate 180 (vec08u *dst, vec08u *src, int bw, int bh, int sstr)

180-degree rotation.

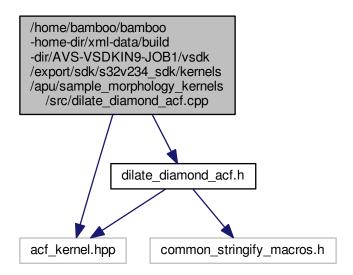
7.119.1 Detailed Description

180-degree rotation implementation for APEX

7.120 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_acf.cpp File
Reference

```
#include "acf_kernel.hpp"
#include "dilate_diamond_acf.h"
```

Include dependency graph for dilate_diamond_acf.cpp:



Functions

7.121 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.cpp File Reference

dilate_diamond implementation for APEX

7.121.1 Detailed Description

dilate_diamond implementation for APEX

Author

Igor Aleksandrowicz

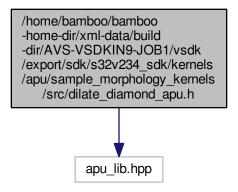
Version

Date

7.122 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_morphology_kernels/src/dilate_diamond_apu.h File Reference

Image diamond dilation implementation for APEX.

```
#include "apu_lib.hpp"
Include dependency graph for dilate_diamond_apu.h:
```



Functions

• void dilate_diamond (vec08u *dst, vec08u *src, int dstr, int sstr, int bw, int bh)

5x5 diamond dilation.

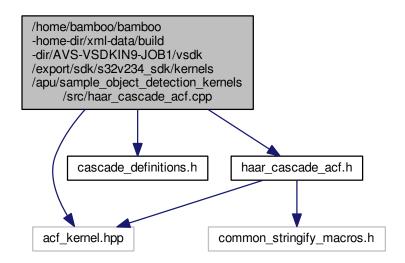
7.122.1 Detailed Description

Image diamond dilation implementation for APEX.

7.123 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "cascade_definitions.h"
#include "haar_cascade_acf.h"
```

Include dependency graph for haar_cascade_acf.cpp:



Functions

NTEGRAL_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), _e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("INPUT_INTEGRAL_IMAGE_SQUARE ← D"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), __e0_size(1, 1), _ek_size(1, 1)), __port(__index(2), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_SCL_OUT_ST↔ ATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d16u), e0 size(1, 1), ek size(1, 1), port $(\leftarrow$ __index(3), __identifier("WINDOW_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), __~ $spatial_dep(0,\ 0,\ 0,\ 0),\ __e0_data_type(d32u),\ __e0_size(1,\ 1),\ __ek_size(6+1,\ 32)),\ __port(__index(4),\ \hookleftarrow 1,\ 0)$ $\underline{\hspace{0.1cm}} \text{identifier} (\text{"WINDOW}_\text{BUFFER}_\text{SQUARED"}), \ \underline{\hspace{0.1cm}} \text{attributes} (\text{ACF}_\text{ATTR}_\text{VEC}_\text{OUT}_\text{STATIC}_\text{FIXED}), \ \underline{\hspace{0.1cm}} \leftarrow$ spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(6+1, 32)), __port(__index(5), dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(2, 1)), __port(__index(6), __identifier("I← NPUT CASCADE FEATURES"), attributes(ACF ATTR SCL IN STATIC FIXED), spatial dep(0, 0, $0, \ 0), \ __e0_data_type(d08u), \ __e0_size(1, \ 1), \ __ek_size(19800, \ 1)), \ __port(__index(7), \ __identifier("IN \leftarrow 1000 \ 1000$ PUT_CASCADE_STAGES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), _e0_data_type(d08u), __e0_size(1, 1), __ek_size(70, 1)), __port(_index(8), __identifier("INPUT_PIXE↔ $L_SHIFTS"), \ __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), \ __spatial_dep(0, \ 0, \ 0), \ __e0_data_{\leftarrow} \\$ type(d08u), __e0_size(1, 1), __ek_size(32, 1)), __port(__index(9), __identifier("INPUT_PIXEL_OFFSETS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __ e0_size(1, 1), __ek_size(32, 1)), __port(_index(10), __identifier("OUTPUT"), __attributes(ACF_ATTR_V← EC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))

7.124 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.cpp File Reference

Haar cascade implementation for APEX.

7.124.1 Detailed Description

Haar cascade implementation for APEX.

Author

Igor Aleksandrowicz

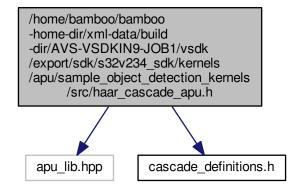
Version

Date

7.125 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_object_detection_kernels/src/haar_cascade_apu.h File
Reference

Object detection based on Haar-like features implementation for APEX.

```
#include <apu_lib.hpp>
#include "cascade_definitions.h"
Include dependency graph for haar_cascade_apu.h:
```



Classes

- struct APEX HaarCascadeFeature
- struct APEX_HaarCascadeStage

Typedefs

- typedef vec16u **FEATURE_FIXED_POINT_TYPE**
- typedef vec16u STAGE_FIXED_POINT_TYPE

Reference 343 Functions

void haar_cascade (vec08u *apOut, vec32u *apInII, vec32u *apInII2, int aOutStride, int aInStride, int a
 —
 TileWidth, int aTileHeight, int16u aLineIndex, vec32u *apWindowBuffer, vec32u *apWindowBuffer2, const
 APEX_HaarCascadeFeature *apcFeatures, int aStageCount, const APEX_HaarCascadeStage *apcStages,
 const int08u *apcXshifts, const int08u *apcXoffsets)

Haar object detection.

Variables

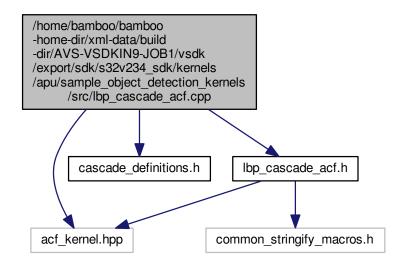
- const int **FEATURE_FRACTIONAL_BITS** = 13
- const int FEATURE FIXED POINT MULTIPLIER = (1 << FEATURE FRACTIONAL BITS)
- const int STAGE_FRACTIONAL_BITS = 10
- const int STAGE FIXED POINT MULTIPLIER = (1 << STAGE FRACTIONAL BITS)
- const int32u featureFixedCoefficientSqrt = 91
- const int32s invWindowAreaScalar = 25

7.125.1 Detailed Description

Object detection based on Haar-like features implementation for APEX.

7.126 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_acf.cpp File
Reference

```
#include "acf_kernel.hpp"
#include "cascade_definitions.h"
#include "lbp_cascade_acf.h"
Include dependency graph for lbp_cascade_acf.cpp:
```



Functions

 KERNEL_INFO apu_lbp_cascade (" apu_lbp_cascade ", 9, __port(__index(0), __identifier("INPUT_INTE--GRAL_IMAGE"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 0, 1, 0), __e0_data_type(d32u), __ e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("LINE_INDEX"), __attributes(ACF_ATTR_S ← CL OUT STATIC FIXED), spatial dep(0, 0, 0, 0), e0 data type(d16u), e0 size(1, 1), ek size(1, 1)), __port(__index(2), __identifier("WINDOW_BUFFER"), __attributes(ACF_ATTR_VEC_OUT_STATIC_ ~ FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(20+1, 32)), __port(← _index(3), __identifier("INPUT_CASCADE_SIZES_AND_SKIP"), __attributes(ACF_ATTR_SCL_IN_STA- $TIC_FIXED), __spatial_dep(0, \, 0, \, 0, \, 0), \, __e0_data_type(d16u), \, __e0_size(1, \, 1), \, __ek_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __ex_size(3, \, 1)), \, __port(\leftarrow 1, \, 1), \, __port(\leftarrow 1, \, 1),$ __index(4), __identifier("INPUT_CASCADE_FEATURES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FI-XED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(10000, 1)), __port(_\iff
_ _index(5), __identifier("INPUT_CASCADE_STAGES"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(200, 1)), __port(__index(6), _ <-_identifier("INPUT_PIXEL_SHIFTS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(64, 1)), __port(_ index(7), __identifier("INP← UT_PIXEL_OFFSETS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), _ c _e0_data_type(d08u), __e0_size(1, 1), __ek_size(64, 1)), __port(__index(8), __identifier("OUTPUT"), __ attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ ek size(1, 1)))

7.127 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.cpp File
Reference

LBP cascade implementation for APEX.

7.127.1 Detailed Description

LBP cascade implementation for APEX.

Author

Igor Aleksandrowicz

Version

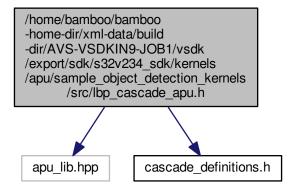
Date

7.128 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_object_detection_kernels/src/lbp_cascade_apu.h File
Reference

Object detection based on LBP features implementation for APEX.

```
#include <apu_lib.hpp>
#include "cascade_definitions.h"
```

Include dependency graph for lbp_cascade_apu.h:



Classes

- struct APEX_lbpFeature
- struct APEX_lbpStage

Typedefs

- typedef vec32s STAGE FIXED POINT TYPE
- typedef int32s STAGE_FIXED_POINT_TYPE_SCALAR

Functions

void lbp_cascade (vec08u *apOut, vec32u *apInII, int aOutStride, int aInStride, int aTileWidth, int aTileHeight, int16u aLineIndex, vec32u *apWindowBuffer, const APEX_lbpFeature *apcFeatures, int aStageCount, const APEX_lbpStage *apcStages, const int08u *apcXshifts, const int08u *apcXoffsets, int skipOdd)

Local Binary Pattern object detection.

Variables

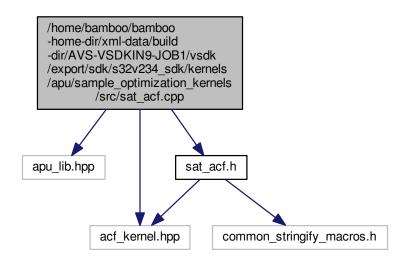
- const int STAGE_FRACTIONAL_BITS = 28
- const int **STAGE_FIXED_POINT_MULTIPLIER** = (1 << STAGE_FRACTIONAL_BITS)

7.128.1 Detailed Description

Object detection based on LBP features implementation for APEX.

7.129 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 sdk/kernels/apu/sample optimization kernels/src/sat acf.cpp File Reference

```
#include "apu_lib.hpp"
#include "acf_kernel.hpp"
#include "sat_acf.h"
Include dependency graph for sat acf.cpp:
```



Functions

KERNEL_INFO apu_sat (" apu_sat ", 3, __port(__index(0), __identifier("INPUT"), __attributes(ACF_ATTR \cup VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)), __port(_\cup index(1), __identifier("OUTPUT"), __attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_\cup data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(2), __identifier("OUTPUT_ROW"), __\cup attributes(ACF_ATTR_VEC_OUT_STATIC), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

7.130 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_optimization_kernels/src/sat_apu.cpp File Reference

sat box filter implementation for APEX

7.130.1 Detailed Description

sat_box_filter implementation for APEX

Author

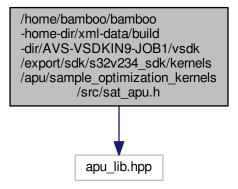
Igor Aleksandrowicz

Date

7.131 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 sdk/kernels/apu/sample optimization kernels/src/sat apu.h File Reference

Summed area table implementation for APEX.

```
#include "apu_lib.hpp"
Include dependency graph for sat_apu.h:
```



Functions

void sat32 (vec32u *apDest, vec32u *apPrevRow, const vec08u *apcSrc, int aSourceStride, int a
 —
 DestinationStride, int aBlockWidth, int aBlockHeight, int08u aFirstTile)

Summed area table.

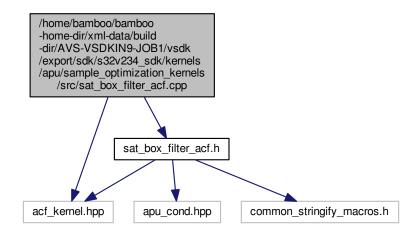
7.131.1 Detailed Description

Summed area table implementation for APEX.

7.132 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "sat_box_filter_acf.h"
```

Include dependency graph for sat_box_filter_acf.cpp:



Functions

KERNEL_INFO apu_sat_box_filter (" apu_sat_box_filter ", 2, __port(_index(0), __identifier("INPUT"), _
 _ attributes(ACF_ATTR_VEC_IN), __spatial_dep(BOX_SIZE+1, BOX_SIZE, BOX_SIZE+1, BOX_SIZE),
 _ e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)), __port(__index(1), __identifier("OUTPUT"), __
 attributes(ACF_ATTR_VEC_OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __
 ek_size(1, 1)))

Variables

- const int BOX_SIZE
- · const int BOX_AREA
- 7.133 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
 _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.cpp File
 Reference

sat_box_filter implementation for APEX

7.133.1 Detailed Description

sat_box_filter implementation for APEX

Author

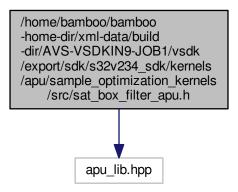
Igor Aleksandrowicz

Version

7.134 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_optimization_kernels/src/sat_box_filter_apu.h File Reference

Box filter using SAT implementation for APEX.

#include "apu_lib.hpp"
Include dependency graph for sat_box_filter_apu.h:



Functions

void sat_box_filter_impl (vec08u *apDest, const vec32u *apcSrc, int aBlockWidth, int aBlockHeight, int a
 — SourceStride, int aDestStride)

Sum of values over one patch the input image is a SAT image (i.e. integral computed with the sat32() function.

Variables

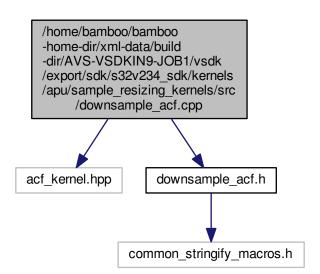
- · const int BOX_SIZE
- const int BOX_AREA

7.134.1 Detailed Description

Box filter using SAT implementation for APEX.

7.135 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/downsample_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "downsample_acf.h"
Include dependency graph for downsample_acf.cpp:
```



Functions

- KERNEL_INFO apu_downsample (" apu_downsample ", 2, __port(__index(0), __identifier("INPUT_0"), __
 attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_
 size(2, 2)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_
 dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_downsample_16u (" apu_downsample_16u ", 2, __port(_index(0), __identifier("INP
 UT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1,
 1), __ek_size(2, 2)), __port(_index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT),
 __spatial_dep(0, 0, 0, 0), __e0_data_type(d16u), __e0_size(1, 1), __ek_size(1, 1)))
- KERNEL_INFO apu_downsample_gauss (" apu_downsample_gauss ", 2, __port(__index(0), __identifier("I ← NPUT_0"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(2, 2)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_ ← OUT), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(1, 1)))
- 7.136 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.cpp File Reference

downsample implementation for APEX

7.136.1 Detailed Description

downsample implementation for APEX

Author

Igor Aleksandrowicz

Version

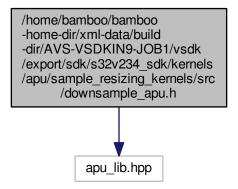
Date

7.137 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/downsample_apu.h File Reference

Image downsample implementation for APEX.

#include "apu_lib.hpp"

Include dependency graph for downsample apu.h:



Functions

void downsample (vec08u *apDest, const vec08u *apcSrc, int aOutBlockWidth, int aOutBlockHeight, int a
 —
 InBlockStride, int aOutBlockStride)

x2 downsampling.

void downsample_16u (vec16u *apDest, const vec16u *apcSrc, int alnBlockWidth, int alnBlockHeight, int aOutBlockWidth, int aOutBlockHeight, int alnBlockStride, int aOutBlockStride)

x2 downsampling, 16-bit.

void downsample_gauss (vec08u *apDest, const vec08u *apcSrc, int32s aOutBlockWidth, int32s aOut
 BlockHeight, int32s aInBlockStride, int32s aOutBlockStride)

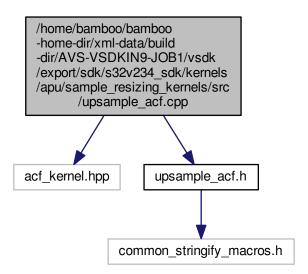
x2 downsampling using Gaussian blur.

7.137.1 Detailed Description

Image downsample implementation for APEX.

7.138 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/upsample_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "upsample_acf.h"
Include dependency graph for upsample_acf.cpp:
```



Functions

• KERNEL_INFO apu_upsample (" apu_upsample ", 2, __port(__index(0), __identifier("INPUT_0"), __ \hookleftarrow attributes(ACF_ATTR_VEC_IN), __spatial_dep(1, 1, 1, 1), __e0_data_type(d08u), __e0_size(1, 1), __ek_ \hookleftarrow size(1, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_OUT), __spatial_ \hookleftarrow dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_size(1, 1), __ek_size(2, 2)))

7.139 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.cpp File Reference

upsample implementation for APEX

7.139.1 Detailed Description

upsample implementation for APEX

Igor Aleksandrowicz

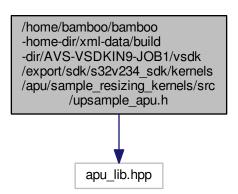
Version

Date

7.140 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_resizing_kernels/src/upsample_apu.h File Reference

Image upsample implementation for APEX.

#include "apu_lib.hpp"
Include dependency graph for upsample_apu.h:



Functions

void upsample (vec08u *apDest, const vec08u *apcSrc, int alnBlockWidth, int alnBlockHeight, int alnBlockWidth, int alnBlockHeight, int alnBlockWidth, int alnBlockHeight, int alnBlockWidth, int alnBlockHeight, int

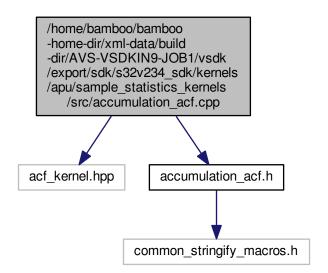
x2 upsampling.

7.140.1 Detailed Description

Image upsample implementation for APEX.

7.141 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/accumulation_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "accumulation_acf.h"
Include dependency graph for accumulation acf.cpp:
```



Macros

- #define ACCUM TILE SIZE X 10
- #define ACCUM_TILE_SIZE_Y 10

Functions

• KERNEL_INFO apu_accumulation (" apu_accumulation ", 6, __port(_index(0), __identifier("INPUT_Img"), __attributes(ACF_ATTR_VEC_IN), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ ⇔ ek_size(10, 10)), __port(_index(1), __identifier("Output_Img"), __attributes(ACF_ATTR_VEC_OUT), __ ⇔ spatial_dep(0, 0, 0, 0), __e0_data_type(d32s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(2), __ ⇔ identifier("ACCUM_XOFFS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(3), __identifier("ACCUM_YOF ⇔ FS"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)), __port(_index(4), __identifier("ACCUM_XWIDTH"), __attributes(ACF_⇔ ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1), __ek_size(1, 1), __ek_size(1, 1)), __static_butes(ACF_ATTR_SCL_IN_STAH ⇔ TIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d16s), __e0_size(1, 1), __ek_size(1, 1)))

7.142 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234—sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.cpp File

Reference 7.142 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.cpp File Reference

Builds the sum of all elements of a chunk and writes out a vector of sum values.

7.142.1 Detailed Description

Builds the sum of all elements of a chunk and writes out a vector of sum values.

Author

Anca Dima

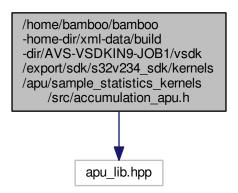
Version

Date

7.143 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/accumulation_apu.h File Reference

accumulation of values from a chunk

#include "apu_lib.hpp"
Include dependency graph for accumulation_apu.h:



Functions

void accumulation_in32s_filter (vec32s *dst, vec32s *srcA, int16s sstr, int16s xOffs, int16s yOffs, int16s x←
 AccWidth, int16s yAccHeight)

Accumulates all values in a chunk (signed 32bit).

void accumulation_in32u_filter (vec32u *lpvOut, vec32u *lpvIn, int16s strideWidth, int16s xOffs, int16s yOffs, int16s xAccWidth, int16s yAccHeight)

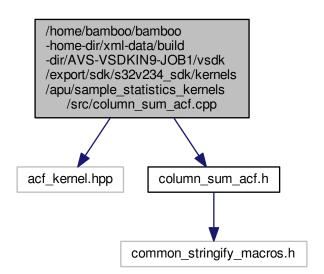
Accumulates all values in a chunk (unsigned 32bit).

7.143.1 Detailed Description

accumulation of values from a chunk

7.144 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/column_sum_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "column_sum_acf.h"
Include dependency graph for column sum acf.cpp:
```



Functions

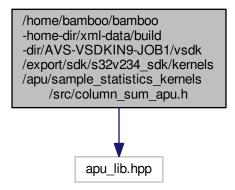
KERNEL_INFO apu_columns_sum (" apu_columns_sum ", 6, __port(__index(0), __identifier("INPUT_ ← 0"), __attributes(ACF_ATTR_VEC_IN_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d08u), __e0_ ← size(1, 1), __ek_size(4, 10)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_VEC_COUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(2), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_VEC_OUT_STATIC_FIXED), ← __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(3), ← __dentifier("INPUT_1"), __attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __← attributes(ACF_ATTR_SCL_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_ ← size(1, 1), __ek_size(1, 1)), __port(__index(5), __identifier("INPUT_3"), __attributes(ACF_ATTR_SCL_IN ← STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(1, 1)))

7.145 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v234--__sdk/kernels/apu/sample_statistics_kernels/src/column_sum_apu.h File

Reference
7.145 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_statistics_kernels/src/column_sum_apu.h File Reference
ence

cumputing sum of columns of image for APEX ldw_v2 demo

```
#include "apu_lib.hpp"
Include dependency graph for column_sum_apu.h:
```



Functions

• void column_sum (vec08u *lpvIn0, vec32u *lpvOutDown, vec32u *lpvOutUp, bool isFirstTile, int IStrideIn0, int chunkWidth, int chunkHeight, int outChunkWidth, int priorityDown, int priorityUp, int indexOfTileStart)

Elementwise unsigned 8bit addition => unsigned 16bit.

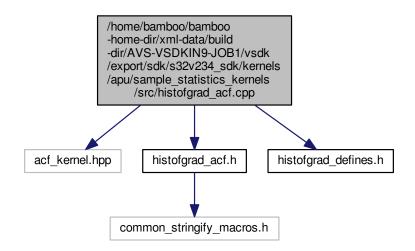
7.145.1 Detailed Description

cumputing sum of columns of image for APEX ldw_v2 demo

7.146 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "histofgrad_acf.h"
#include "histofgrad_defines.h"
```

Include dependency graph for histofgrad_acf.cpp:



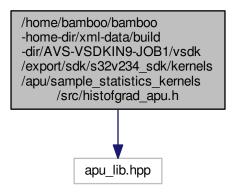
Functions

7.147 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/histofgrad_apu.h File Reference

histogram of gradients computation for APEX

#include "apu_lib.hpp"

Include dependency graph for histofgrad_apu.h:



Functions

• void hog (vec08s *lpvInGradX, vec08s *lpvInGradY, vec16u *lpvOut, vec32u *lpvOutBINorm, int IStrideIn, int chunkWidth, int chunkHeight, int IStrideOut)

Elementwise unsigned 8bit addition => unsigned 16bit.

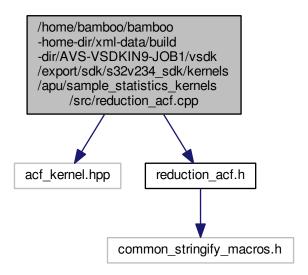
7.147.1 Detailed Description

histogram of gradients computation for APEX

7.148 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_statistics_kernels/src/reduction_acf.cpp File Reference

```
#include "acf_kernel.hpp"
#include "reduction_acf.h"
```

Include dependency graph for reduction_acf.cpp:



Functions

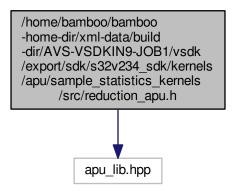
• KERNEL_INFO apu_reduction (" apu_reduction ", 2, __port(__index(0), __identifier("INPUT_0"), __ attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_ size(1, 1), __ek_size(256, 1)), __port(__index(1), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SC L_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(256, 1)))

7.149 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/sample_statistics_kernels/src/reduction_apu.h File Reference

reduction of a vector/image implementation for APEX

#include "apu_lib.hpp"

Include dependency graph for reduction_apu.h:



Functions

• void reduc (vec32u *lpvIn0, int32s *lpOut0, bool isLastTile, int16s IFirstCuld, int16s ITileWidthInChunks, int IChunkWidth, int IChunkHeight, int IChunkSpanIn0, int IChunkSpanOut0)

Elementwise unsigned 8bit addition => unsigned 16bit.

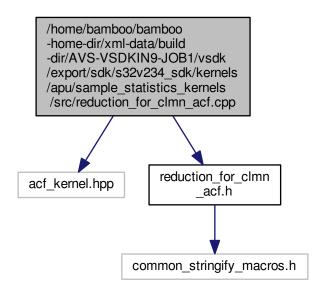
7.149.1 Detailed Description

reduction of a vector/image implementation for APEX

7.150 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
_sdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_acf.cpp File
Reference

```
#include "acf_kernel.hpp"
#include "reduction_for_clmn_acf.h"
```

Include dependency graph for reduction_for_clmn_acf.cpp:



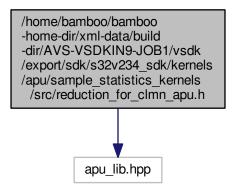
Functions

- KERNEL_INFO apu_reduction_for_clmn (" apu_reduction_for_clmn ", 4, __port(__index(0), __identifier("I← NPUT_0"), __attributes(ACF_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_← type(d32u), __e0_size(1, 1), __ek_size(4, 1)), __port(__index(1), __identifier("INPUT_1"), __attributes(AC← F_ATTR_VEC_IN_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __← ek_size(4, 1)), __port(__index(2), __identifier("OUTPUT_0"), __attributes(ACF_ATTR_SCL_OUT_STATIC← __FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)), __port(_ ← index(3), __identifier("OUTPUT_1"), __attributes(ACF_ATTR_SCL_OUT_STATIC_FIXED), __spatial_dep(0, 0, 0, 0), __e0_data_type(d32u), __e0_size(1, 1), __ek_size(192, 1)))
- 7.151 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23
 _sdk/kernels/apu/sample_statistics_kernels/src/reduction_for_clmn_apu.h File
 Reference

reduction of a vector/image implementation for APEX ldw_v2 demo

#include "apu_lib.hpp"

Include dependency graph for reduction_for_clmn_apu.h:



Functions

• void reduc (vec32u *lpvIn0, vec32u *lpvIn1, int32s *lpOut0, int32s *lpOut1, bool isLastTile, int16s lFirstCuld, int16s lTileWidthInChunks, int lChunkWidth, int lChunkHeight, int lChunkSpanIn0, int lChunkSpanOut0)

Elementwise unsigned 8bit addition => unsigned 16bit.

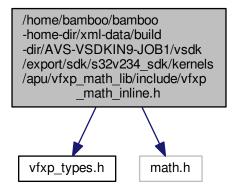
7.151.1 Detailed Description

reduction of a vector/image implementation for APEX ldw_v2 demo

7.152 /home/bamboo/bamboo-home-dir/xml-data/build-dir/AVS-VSDKIN9-JOB1/vsdk/export/sdk/s32v23 _sdk/kernels/apu/vfxp_math_lib/include/vfxp_math_inline.h File Reference

```
#include "vfxp_types.h"
#include "math.h"
```

Include dependency graph for vfxp_math_inline.h:



Macros

• #define ALWAYS_INLINE static inline

Index

Accumulation, 192	upsample, 187
add	lower
Addition, 28	Comparison, 91
Addition, 27	Companison, C1
add, 28	mark
disparity, 35	Marking on images, 108
Arithmetic, 26	Marking on images, 108
binarize	mark, 108
	mask
Saturation, 156	Saturation, 157
Binary descriptor matching, 95	Match
Match, 96	Binary descriptor matching, 96
Bit width conversion, 98	max
Color conversion, 100	Maximum, 75
	Maximum, 75
Comparison, 77, 78	max, 75
lower, 91	Median filtering, 141
Conversion, 97	
Correlation, 124	Morphology, 166
Difference, 36	Multiplication, 51
Dilation, 167	nms3x3
	Fast, 21
disparity Addition 25	•
Addition, 35	Non-maximum suppression, 151
Display, 106	Object detection, 169
Division, 44	of gradients, 197
downsample Image downsampling, 185	hist, 198
	hog, 198
Fact 01	Optimization, 178
Fast, 21	Optimization, 176
nms3x3, 21	reduc
Fast9 feature detection, 111	Reduction, 200
Feature detection, 110	Reduction, 199
Filtering, 120	reduc, 200
filtering, 138	Remap bilinear, 159
Causaian filtaring 145	Remap nearest, 162
Gaussian filtering, 145	Resizing, 183
Geometry, 158	Rotate image by 180 deg, 164
Haar cascade object detection, 170	Tiotate image by 100 deg, 104
Harris corner detection, 113	sat32
hist	Summed area table, 179
	Saturation, 155
of gradients, 198	binarize, 156
Histofgrad, 196	mask, 157
hog	Sobel Filtering, 143
of gradients, 198	Square, 69
Image downsampling, 184	Statistics, 189
	Sum of Absolute Differences, 117
downsample, 185	
Image gradient, 147 Image upsampling, 187	Summed area table, 179 sat32, 179
imaue ubsambimu. 10/	Saloc. 1/9

366 INDEX

upsample

Image upsampling, 187