



# CV180X & CV181X ISP 开发使用手册

Version: 1.3.2

Release date: 2023-02-06

©2022 北京晶视智能科技有限公司  
本文件所含信息归北京晶视智能科技有限公司所有。  
未经授权，严禁全部或部分复制或披露该等信息。

# 目录

<b>1</b>	<b>声明</b>	<b>2</b>
<b>2</b>	<b>概述</b>	<b>3</b>
2.1	概述 . . . . .	3
2.2	功能描述 . . . . .	3
2.3	架构 . . . . .	4
2.4	开发模式 . . . . .	4
2.5	内部流程 . . . . .	5
2.6	软件流程 . . . . .	6
<b>3</b>	<b>系统控制</b>	<b>8</b>
3.1	功能概述 . . . . .	8
3.2	API 参考 . . . . .	8
3.2.1	CVI_ISP_MemInit . . . . .	9
3.2.2	CVI_ISP_Init . . . . .	10
3.2.3	CVI_ISP_Run . . . . .	11
3.2.4	CVI_ISP_RunOnce . . . . .	11
3.2.5	CVI_ISP_Exit . . . . .	12
3.2.6	CVI_ISP_SetPubAttr . . . . .	13
3.2.7	CVI_ISP_GetPubAttr . . . . .	14
3.2.8	CVI_ISP_SetFMWState . . . . .	14
3.2.9	CVI_ISP_GetFMWState . . . . .	15
3.2.10	CVI_ISP_SetModuleControl . . . . .	16
3.2.11	CVI_ISP_GetModuleControl . . . . .	17
3.2.12	CVI_ISP_GetVDTimeOut . . . . .	17
3.2.13	CVI_ISP_SensorRegCallBack . . . . .	18
3.2.14	CVI_ISP_SensorUnRegCallBack . . . . .	19
3.2.15	CVI_ISP_AELibRegCallBack . . . . .	20
3.2.16	CVI_ISP_AELibUnRegCallBack . . . . .	21
3.2.17	CVI_ISP_AWBLibRegCallBack . . . . .	22
3.2.18	CVI_ISP_AWBLibUnRegCallBack . . . . .	23
3.2.19	CVI_ISP_SetBindAttr . . . . .	24
3.2.20	CVI_ISP_GetBindAttr . . . . .	25
3.2.21	CVI_ISP_SetCtrlParam . . . . .	26
3.2.22	CVI_ISP_GetCtrlParam . . . . .	27
3.2.23	CVI_ISP_SetModParam . . . . .	27
3.2.24	CVI_ISP_GetModParam . . . . .	28
3.2.25	CVI_BIN_SetBinName . . . . .	29
3.2.26	CVI_BIN_GetBinName . . . . .	29
3.2.27	CVI_BIN_GetBinExtraAttr . . . . .	29
3.2.28	CVI_BIN_GetBinTotalLen . . . . .	29

3.2.29	CVI_BIN_ExportBinData	29
3.2.30	CVI_BIN_ImportBinData	30
3.2.31	CVI_BIN_SaveParamToBin	30
3.2.32	CVI_BIN_LoadParamFromBin	30
3.2.33	CVI_ISP_IrAutoRunOnce	30
3.2.34	CVI_ISP_SetSmartInfo	31
3.2.35	CVI_ISP_GetSmartInfo	32
3.3	数据类型	33
3.3.1	RECT_S	33
3.3.2	SIZE_S	34
3.3.3	ISP_BAYER_FORMAT_E	34
3.3.4	WDR_MODE_E	35
3.3.5	ISP_PUB_ATTR_S	36
3.3.6	ISP_FMW_STATE_E	37
3.3.7	ISP_MODULE_CTRL_U	38
3.3.8	ISP_VD_TYPE_E	39
3.3.9	ISP_SNS_ATTR_INFO_S	40
3.3.10	ALG_LIB_S	40
3.3.11	ISP_AE_REGISTER_S	41
3.3.12	ISP_AE_EXP_FUNC_S	41
3.3.13	ISP_AE_PARAM_S	42
3.3.14	ISP_SMART_INFO_S	43
3.3.15	ISP_AE_INFO_S	44
3.3.16	ISP_AE_RESULT_S	45
3.3.17	ISP_SENSOR_EXP_FUNC_S	47
3.3.18	ISP_SENSOR_REGISTER_S	48
3.3.19	ISP_AWB_EXP_FUNC_S	48
3.3.20	ISP_AWB_REGISTER_S	49
3.3.21	ISP_AWB_PARAM_S	49
3.3.22	ISP_AWB_STAT_1_S	50
3.3.23	ISP_AWB_STAT_RESULT_S	51
3.3.24	ISP_AWB_INFO_S	51
3.3.25	ISP_SMART_ROI_S	52
3.3.26	ISP_AWB_RAW_STAT_ATTR_S	53
3.3.27	ISP_AWB_RESULT_S	54
3.3.28	ISP_BIND_ATTR_S	55
3.3.29	ISP_CTRL_PARAM_S	55
3.3.30	ISP_MOD_PARAM_S	56
3.3.31	ISP_IR_AUTO_ATTR_S	57
4	<b>AE</b>	<b>59</b>
4.1	概述	59
4.2	重要概念	60
4.3	功能描述	60
4.4	API 参考	62
4.4.1	AE 库接口	62
4.4.1.1	CVI_AE_Register	62
4.4.1.2	CVI_AE_UnRegister	63
4.4.1.3	CVI_AE_SensorRegCallBack	63
4.4.1.4	CVI_AE_SensorUnRegCallBack	65

4.4.2	AE 控制模块	66
4.4.2.1	CVI_ISP_SetExposureAttr	66
4.4.2.2	CVI_ISP_GetExposureAttr	71
4.4.2.3	CVI_ISP_SetWDRExposureAttr	71
4.4.2.4	CVI_ISP_GetWDRExposureAttr	73
4.4.2.5	CVI_ISP_SetAERouteAttr	74
4.4.2.6	CVI_ISP_GetAERouteAttr	76
4.4.2.7	CVI_ISP_SetAERouteAttrEx	76
4.4.2.8	CVI_ISP_GetAERouteAttrEx	77
4.4.2.9	CVI_ISP_QueryExposureInfo	78
4.4.2.10	CVI_ISP_SetAntiFlicker	79
4.4.2.11	CVI_ISP_GetAntiFlicker	80
4.4.2.12	CVI_ISP_QueryFps	81
4.4.2.13	CVI_ISP_GetCurrentLvX100	82
4.4.2.14	CVI_ISP_SetSmartExposureAttr	83
4.4.2.15	CVI_ISP_GetSmartExposureAttr	84
4.4.2.16	CVI_ISP_SetAERouteSFAttr	85
4.4.2.17	CVI_ISP_GetAERouteSFAttr	86
4.4.2.18	CVI_ISP_SetAERouteSFAttrEx	86
4.4.2.19	CVI_ISP_GetAERouteSFAttrEx	87
4.4.3	光圈控制模块	88
4.4.3.1	CVI_ISP_SetIrisAttr	88
4.4.3.2	CVI_ISP_GetIrisAttr	89
4.4.3.3	CVI_ISP_SetDcirisAttr	90
4.4.3.4	CVI_ISP_GetDcirisAttr	91
4.4.3.5	CVI_ISP_SetPirisAttr	91
4.4.3.6	CVI_ISP_GetPirisAttr	92
4.5	数据类型	93
4.5.1	Register	93
4.5.1.1	AE_SENSOR_REGISTER_S	93
4.5.1.2	AE_SENSOR_EXP_FUNC_S	94
4.5.1.3	AE_SENSOR_DEFAULT_S	95
4.5.1.4	AE_ACCURACY_E	97
4.5.1.5	AE_ACCURACY_S	98
4.5.2	AE	98
4.5.2.1	ISP_AE_MODE_E	99
4.5.2.2	ISP_AE_STRATEGY_E	99
4.5.2.3	ISP_AE_DELAY_S	100
4.5.2.4	ISP_AE_RANGE_S	101
4.5.2.5	ISP_ANTIFLICKER_MODE_E	101
4.5.2.6	ISP_ANTIFLICKER_S	102
4.5.2.7	ISP_SUBFLICKER_S	103
4.5.2.8	ISP_FSWDR_MODE_E	103
4.5.2.9	ISP_AE_GAIN_TYPE_E	104
4.5.2.10	ISP_AE_ATTR_S	105
4.5.2.11	ISP_ME_ATTR_S	110
4.5.2.12	ISP_EXPOSURE_ATTR_S	112
4.5.2.13	ISP_WDR_EXPOSURE_ATTR_S	113
4.5.2.14	ISP_AE_ROUTE_NODE_S	115
4.5.2.15	ISP_AE_ROUTE_S	116

4.5.2.16	ISP_AE_ROUTE_EX_NODE_S	116
4.5.2.17	ISP_AE_ROUTE_EX_S	117
4.5.2.18	ISP_EXP_INFO_S	118
4.5.2.19	ISP_SMART_EXPOSURE_ATTR_S	120
4.5.3	IRIS	122
4.5.3.1	ISP_IRIS_STATUS_E	122
4.5.3.2	ISP_IRIS_TYPE_E	123
4.5.3.3	ISP_IRIS_F_NO_E	123
4.5.3.4	ISP_MI_ATTR_S	124
4.5.3.5	ISP_DCIRIS_ATTR_S	125
4.5.3.6	ISP_PIRIS_ATTR_S	127
4.5.3.7	ISP_IRIS_ATTR_S	127
<b>5</b>	<b>AWB</b>	<b>129</b>
5.1	概述	129
5.2	重要概念	129
5.3	功能描述	129
5.4	API 参考	130
5.4.1	AWB 库接口	130
5.4.1.1	CVI_AWB_Register	130
5.4.1.2	CVI_AWB_UnRegister	131
5.4.1.3	CVI_AWB_SensorRegCallBack	132
5.4.1.4	CVI_AWB_SensorUnRegCallBack	133
5.4.2	AWB 控制模块	134
5.4.2.1	CVI_ISP_SetWBAttr	134
5.4.2.2	CVI_ISP_GetWBAttr	135
5.4.2.3	CVI_ISP_SetAWBAttrEx	136
5.4.2.4	CVI_ISP_GetAWBAttrEx	137
5.4.2.5	CVI_ISP_QueryWBInfo	138
5.4.2.6	CVI_ISP_SetAWBLogPath	138
5.4.2.7	CVI_ISP_SetAWBLogName	139
5.4.2.8	CVI_ISP_GetGrayWorldAwbInfo	140
5.5	数据类型	141
5.5.1	Register	141
5.5.2	WB	141
5.5.2.1	ISP_MWB_ATTR_S	142
5.5.2.2	ISP_AWB_CT_LIMIT_ATTR_S	142
5.5.2.3	ISP_AWB_ALG_E	143
5.5.2.4	ISP_AWB_ATTR_S	144
5.5.2.5	ISP_AWB_ALG_TYPE_E	146
5.5.2.6	ISP_AWB_CBCR_TRACK_ATTR_S	146
5.5.2.7	ISP_AWB_LUM_HISTGRAM_ATTR_S	147
5.5.2.8	ISP_WB_ATTR_S	148
5.5.2.9	ISP_AWB_ATTR_EX_S	148
5.5.2.10	ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S	150
5.5.2.11	ISP_AWB_IN_OUT_ATTR_S	151
5.5.2.12	ISP_AWB_MULTI_LS_TYPE_E	152
5.5.2.13	ISP_AWB_INDOOR_OUTDOOR_STATUS_E	153
5.5.2.14	ISP_WB_INFO_S	153

<b>7</b>	<b>BlackLevel</b>	<b>156</b>
7.1	功能描述	156
7.2	API 参考	156
7.2.1	CVI_ISP_SetBlackLevelAttr	156
7.2.2	CVI_ISP_GetBlackLevelAttr	157
7.3	数据类型	158
7.3.1	ISP_BLACK_LEVEL_MANUAL_ATTR_S	158
7.3.2	ISP_BLACK_LEVEL_AUTO_ATTR_S	159
7.3.3	ISP_BLACK_LEVEL_ATTR_S	161
<b>8</b>	<b>DPC</b>	<b>162</b>
8.1	功能描述	162
8.2	API 参考	162
8.2.1	CVI_ISP_SetDPDynamicAttr	163
8.2.2	CVI_ISP_GetDPDynamicAttr	163
8.2.3	CVI_ISP_SetDPCalibrate	164
8.2.4	CVI_ISP_GetDPCalibrate	165
8.2.5	CVI_ISP_SetDPStaticAttr	166
8.2.6	CVI_ISP_GetDPStaticAttr	166
8.3	数据类型	167
8.3.1	ISP_DP_DYNAMIC_MANUAL_ATTR_S	167
8.3.2	ISP_DP_DYNAMIC_AUTO_ATTR_S	169
8.3.3	ISP_DP_DYNAMIC_ATTR_S	170
8.3.4	ISP_DP_CALIB_ATTR_S	171
8.3.5	ISP_DP_STATIC_ATTR_S	173
<b>9</b>	<b>MeshShading</b>	<b>174</b>
9.1	功能描述	174
9.2	API 参考	174
9.2.1	CVI_ISP_SetMeshShadingAttr	174
9.2.2	CVI_ISP_GetMeshShadingAttr	175
9.2.3	CVI_ISP_SetMeshShadingGainLutAttr	176
9.2.4	CVI_ISP_GetMeshShadingGainLutAttr	177
9.3	数据类型	177
9.3.1	ISP_MESH_SHADING_MANUAL_ATTR_S	178
9.3.2	ISP_MESH_SHADING_AUTO_ATTR_S	178
9.3.3	ISP_MESH_SHADING_ATTR_S	179
9.3.4	ISP_MESH_SHADING_GAIN_LUT_S	180
9.3.5	ISP_MESH_SHADING_GAIN_LUT_ATTR_S	180
<b>10</b>	<b>RadialShading</b>	<b>182</b>
10.1	功能描述	182
10.2	API 参考	182
10.2.1	CVI_ISP_SetRadialShadingAttr	182
10.2.2	CVI_ISP_GetRadialShadingAttr	183
10.2.3	CVI_ISP_SetRadialShadingGainLutAttr	184
10.2.4	CVI_ISP_GetRadialShadingGainLutAttr	185
10.3	数据类型	185
10.3.1	ISP_RADIAL_SHADING_MANUAL_ATTR_S	186
10.3.2	ISP_RADIAL_SHADING_AUTO_ATTR_S	186
10.3.3	ISP_RADIAL_SHADING_ATTR_S	187

10.3.4	ISP_RADIAL_SHADING_GAIN_LUT_S	188
10.3.5	ISP_RADIAL_SHADING_GAIN_LUT_ATTR_S	189
<b>11</b>	<b>CCM</b>	<b>191</b>
11.1	功能描述	191
11.2	API 参考	191
11.2.1	CVI_ISP_SetCCMAAttr	191
11.2.2	CVI_ISP_GetCCMAAttr	192
11.2.3	CVI_ISP_SetCCMSaturationAttr	193
11.2.4	CVI_ISP_GetCCMSaturationAttr	193
11.3	数据类型	194
11.3.1	ISP_COLOMATRIX_ATTR_S	195
11.3.2	ISP_CCM_MANUAL_ATTR_S	195
11.3.3	ISP_CCM_AUTO_ATTR_S	196
11.3.4	ISP_CCM_ATTR_S	197
11.3.5	ISP_CCM_SATURATION_MANUAL_ATTR_S	198
11.3.6	ISP_CCM_SATURATION_AUTO_ATTR_S	198
11.3.7	ISP_CCM_SATURATION_ATTR_S	199
<b>12</b>	<b>Noise profile</b>	<b>200</b>
12.1	功能描述	200
12.2	API 参考	200
12.2.1	CVI_ISP_SetNoiseProfileAttr	200
12.2.2	CVI_ISP_GetNoiseProfileAttr	201
12.3	数据类型	202
12.3.1	ISP_CMOS_NOISE_CALIBRATION_S	202
<b>13</b>	<b>BNR</b>	<b>203</b>
13.1	功能描述	203
13.2	API 参考	203
13.2.1	CVI_ISP_SetNRAttr	203
13.2.2	CVI_ISP_GetNRAttr	204
13.2.3	CVI_ISP_SetNRFilterAttr	205
13.2.4	CVI_ISP_GetNRFilterAttr	205
13.3	数据类型	206
13.3.1	ISP_NR_MANUAL_ATTR_S	207
13.3.2	ISP_NR_AUTO_ATTR_S	208
13.3.3	ISP_NR_ATTR_S	209
13.3.4	ISP_NR_FILTER_MANUAL_ATTR_S	210
13.3.5	ISP_NR_FILTER_AUTO_ATTR_S	212
13.3.6	ISP_NR_FILTER_ATTR_S	214
13.3.7	ISP_RLSC_MANUAL_ATTR_S	215
13.3.8	ISP_RLSC_AUTO_ATTR_S	215
13.3.9	ISP_RLSC_ATTR_S	216
<b>14</b>	<b>YNR</b>	<b>217</b>
14.1	功能描述	217
14.2	API 参考	217
14.2.1	CVI_ISP_SetYNRAAttr	217
14.2.2	CVI_ISP_GetYNRAAttr	218
14.2.3	CVI_ISP_SetYNRMotionNRAttr	219

14.2.4	CVI_ISP_GetYNRMotionNRAttr	220
14.2.5	CVI_ISP_SetYNRFilterAttr	220
14.2.6	CVI_ISP_GetYNRFilterAttr	221
14.3	数据类型	222
14.3.1	ISP_YNR_MANUAL_ATTR_S	222
14.3.2	ISP_YNR_AUTO_ATTR_S	223
14.3.3	ISP_YNR_ATTR_S	225
14.3.4	ISP_YNR_MOTION_NR_MANUAL_ATTR_S	227
14.3.5	ISP_YNR_MOTION_NR_AUTO_ATTR_S	227
14.3.6	ISP_YNR_MOTION_NR_ATTR_S	228
14.3.7	ISP_YNR_FILTER_MANUAL_ATTR_S	229
14.3.8	ISP_YNR_FILTER_AUTO_ATTR_S	230
14.3.9	ISP_YNR_FILTER_ATTR_S	231
<b>15</b>	<b>CNR</b>	<b>233</b>
15.1	功能描述	233
15.2	API 参考	233
15.2.1	CVI_ISP_SetCNRAttr	233
15.2.2	CVI_ISP_GetCNRAttr	234
15.2.3	CVI_ISP_SetCNRMotionNRAttr	235
15.2.4	CVI_ISP_GetCNRMotionNRAttr	235
15.3	数据类型	236
15.3.1	ISP_CNR_MANUAL_ATTR_S	237
15.3.2	ISP_CNR_AUTO_ATTR_S	238
15.3.3	ISP_CNR_ATTR_S	239
15.3.4	ISP_CNR_MOTION_NR_MANUAL_ATTR_S	240
15.3.5	ISP_CNR_MOTION_NR_AUTO_ATTR_S	240
15.3.6	ISP_CNR_MOTION_NR_ATTR_S	241
<b>16</b>	<b>TNR</b>	<b>242</b>
16.1	功能描述	242
16.2	API 参考	242
16.2.1	CVI_ISP_SetTNRAttr	242
16.2.2	CVI_ISP_GetTNRAttr	243
16.2.3	CVI_ISP_SetTNRNoiseModelAttr	244
16.2.4	CVI_ISP_GetTNRNoiseModelAttr	245
16.2.5	CVI_ISP_SetTNRNoiseModelAttr	246
16.2.6	CVI_ISP_GetTNRNoiseModelAttr	246
16.2.7	CVI_ISP_SetTNRGhostAttr	247
16.2.8	CVI_ISP_GetTNRGhostAttr	248
16.2.9	CVI_ISP_SetTNRMtPrtAttr	249
16.2.10	CVI_ISP_GetTNRMtPrtAttr	249
16.2.11	CVI_ISP_SetTNRMotionAdaptAttr	250
16.2.12	CVI_ISP_GetTNRMotionAdaptAttr	251
16.3	数据类型	252
16.3.1	ISP_TNR_MANUAL_ATTR_S	252
16.3.2	ISP_TNR_AUTO_ATTR_S	254
16.3.3	ISP_TNR_ATTR_S	255
16.3.4	ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S	257
16.3.5	ISP_TNR_NOISE_MODEL_AUTO_ATTR_S	259
16.3.6	ISP_TNR_NOISE_MODEL_ATTR_S	261



16.3.7	ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S	261
16.3.8	ISP_TNR_LUMA_MOTION_AUTO_ATTR_S	262
16.3.9	ISP_TNR_LUMA_MOTION_ATTR_S	263
16.3.10	ISP_TNR_GHOST_MANUAL_ATTR_S	264
16.3.11	ISP_TNR_GHOST_AUTO_ATTR_S	265
16.3.12	ISP_TNR_GHOST_ATTR_S	266
16.3.13	ISP_TNR_MT_PRT_MANUAL_ATTR_S	266
16.3.14	ISP_TNR_MT_PRT_AUTO_ATTR_S	268
16.3.15	ISP_TNR_MT_PRT_ATTR_S	270
16.3.16	ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S	272
16.3.17	ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S	273
16.3.18	ISP_TNR_MOTION_ADAPT_ATTR_S	274
<b>17</b>	<b>Crosstalk</b>	<b>275</b>
17.1	功能描述	275
17.2	API 参考	275
17.2.1	CVI_ISP_SetCrosstalkAttr	275
17.2.2	CVI_ISP_GetCrosstalkAttr	276
17.3	数据类型	277
17.3.1	ISP_CROSSTALK_MANUAL_ATTR_S	277
17.3.2	ISP_CROSSTALK_AUTO_ATTR_S	277
17.3.3	ISP_CROSSTALK_ATTR_S	278
<b>18</b>	<b>Demosaic</b>	<b>280</b>
18.1	功能描述	280
18.2	API 参考	280
18.2.1	CVI_ISP_SetDemosaicAttr	280
18.2.2	CVI_ISP_GetDemosaicAttr	281
18.2.3	CVI_ISP_SetDemosaicDemoireAttr	282
18.2.4	CVI_ISP_GetDemosaicDemoireAttr	282
18.3	数据类型	283
18.3.1	ISP_DEMOSAIC_MANUAL_ATTR_S	284
18.3.2	ISP_DEMOSAIC_AUTO_ATTR_S	285
18.3.3	ISP_DEMOSAIC_ATTR_S	286
18.3.4	ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S	287
18.3.5	ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S	288
18.3.6	ISP_DEMOSAIC_DEMOIRE_ATTR_S	289
18.3.7	ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S	289
18.3.8	ISP_DEMOSAIC_FILTER_AUTO_ATTR_S	292
18.3.9	ISP_DEMOSAIC_FILTER_ATTR_S	294
<b>19</b>	<b>Sharpen</b>	<b>295</b>
19.1	功能描述	295
19.2	API 参考	295
19.2.1	CVI_ISP_SetSharpenAttr	295
19.2.2	CVI_ISP_GetSharpenAttr	296
19.3	数据类型	297
19.3.1	ISP_SHARPEN_MANUAL_ATTR_S	297
19.3.2	ISP_SHARPEN_AUTO_ATTR_S	299
19.3.3	ISP_SHARPEN_ATTR_S	301

<b>20 PreSharpen</b>	<b>304</b>
20.1 功能描述	304
20.2 API 参考	304
20.2.1 CVI_ISP_SetPreSharpenAttr	304
20.2.2 CVI_ISP_GetPreSharpenAttr	305
20.3 数据类型	306
20.3.1 ISP_PRESHARPEN_MANUAL_ATTR_S	306
20.3.2 ISP_PRESHARPEN_AUTO_ATTR_S	308
20.3.3 ISP_PRESHARPEN_ATTR_S	310
<b>21 Gamma</b>	<b>313</b>
21.1 功能描述	313
21.2 API 参考	313
21.2.1 CVI_ISP_SetGammaAttr	313
21.2.2 CVI_ISP_GetGammaAttr	314
21.2.3 CVI_ISP_GetGammaCurveByType	315
21.2.4 CVI_ISP_SetAutoGammaAttr	316
21.2.5 CVI_ISP_GetAutoGammaAttr	316
21.3 数据类型	317
21.3.1 ISP_GAMMA_CURVE_TYPE_E	317
21.3.2 ISP_GAMMA_ATTR_S	318
21.3.3 ISP_GAMMA_CURVE_ATTR_S	319
21.3.4 ISP_AUTO_GAMMA_ATTR_S	320
<b>22 DCI</b>	<b>321</b>
22.1 功能描述	321
22.2 API 参考	321
22.2.1 CVI_ISP_SetDCIAttr	321
22.2.2 CVI_ISP_GetDCIAttr	322
22.3 数据类型	323
22.3.1 ISP_DCI_MANUAL_ATTR_S	323
22.3.2 ISP_DCI_AUTO_ATTR_S	324
22.3.3 ISP_DCI_ATTR_S	325
<b>23 LDCI</b>	<b>327</b>
23.1 功能描述	327
23.2 API 参考	327
23.2.1 CVI_ISP_SetLDCIAttr	327
23.2.2 CVI_ISP_GetLDCIAttr	328
23.3 数据类型	329
23.3.1 ISP_LDCI_GAUSS_COEF_ATTR_S	329
23.3.2 ISP_LDCI_MANUAL_ATTR_S	330
23.3.3 ISP_LDCI_AUTO_ATTR_S	332
23.3.4 ISP_LDCI_ATTR_S	334
<b>24 Dehaze</b>	<b>335</b>
24.1 功能描述	335
24.2 API 参考	335
24.2.1 CVI_ISP_SetDehazeAttr	335
24.2.2 CVI_ISP_GetDehazeAttr	336
24.3 数据类型	337

24.3.1	ISP_DEHAZE_MANUAL_ATTR_S	337
24.3.2	ISP_DEHAZE_AUTO_ATTR_S	338
24.3.3	ISP_DEHAZE_ATTR_S	338
<b>25</b>	<b>ColorTone</b>	<b>342</b>
25.1	功能描述	342
25.2	API 参考	342
25.2.1	CVI_ISP_SetColorToneAttr	342
25.2.2	CVI_ISP_GetColorToneAttr	343
25.3	数据类型	344
25.3.1	ISP_COLOR_TONE_ATTR_S	344
<b>26</b>	<b>Saturation</b>	<b>345</b>
26.1	功能描述	345
26.2	API 参考	345
26.2.1	CVI_ISP_SetSaturationAttr	345
26.2.2	CVI_ISP_GetSaturationAttr	346
26.3	数据类型	347
26.3.1	ISP_SATURATION_MANUAL_ATTR_S	347
26.3.2	ISP_SATURATION_AUTO_ATTR_S	347
26.3.3	ISP_SATURATION_ATTR_S	348
<b>27</b>	<b>CAC</b>	<b>349</b>
27.1	功能描述	349
27.2	API 参考	349
27.2.1	CVI_ISP_SetCacAttr	349
27.2.2	CVI_ISP_GetCacAttr	350
27.3	数据类型	351
27.3.1	ISP_CAC_MANUAL_ATTR_S	351
27.3.2	ISP_CAC_AUTO_ATTR_S	352
27.3.3	ISP_CAC_ATTR_S	354
<b>28</b>	<b>LCAC</b>	<b>357</b>
28.1	功能描述	357
28.2	API 参考	357
28.2.1	CVI_ISP_SetLCACAttr	357
28.2.2	CVI_ISP_GetLCACAttr	358
28.3	数据类型	359
28.3.1	ISP_LCAC_GAUSS_COEF_ATTR_S	359
28.3.2	ISP_LCAC_MANUAL_ATTR_S	360
28.3.3	ISP_LCAC_AUTO_ATTR_S	361
28.3.4	ISP_LCAC_ATTR_S	362
<b>29</b>	<b>RGBCAC</b>	<b>365</b>
29.1	功能描述	365
29.2	API 参考	365
29.2.1	CVI_ISP_SetRGBCACAttr	365
29.2.2	CVI_ISP_GetRGBCACAttr	366
29.3	数据类型	367
29.3.1	ISP_RGBCAC_MANUAL_ATTR_S	367
29.3.2	ISP_RGBCAC_AUTO_ATTR_S	368

29.3.3	ISP_RGBCAC_ATTR_S	369
<b>30</b>	<b>FSWDR</b>	<b>372</b>
30.1	功能描述	372
30.2	API 参考	372
30.2.1	CVI_ISP_SetFSWDRAttr	372
30.2.2	CVI_ISP_GetFSWDRAttr	373
30.2.3	CVI_ISP_SetWDRExposureAttr	374
30.2.4	CVI_ISP_GetWDRExposureAttr	375
30.3	数据类型	375
30.3.1	ISP_FSWDR_MANUAL_ATTR_S	376
30.3.2	ISP_FSWDR_AUTO_ATTR_S	378
30.3.3	ISP_FSWDR_ATTR_S	380
30.3.4	ISP_WDR_EXPOSURE_ATTR_S	382
<b>31</b>	<b>DRC</b>	<b>385</b>
31.1	功能描述	385
31.2	API 参考	385
31.2.1	CVI_ISP_SetDRCAttr	385
31.2.2	CVI_ISP_GetDRCAttr	386
31.3	数据类型	387
31.3.1	ISP_DRC_MANUAL_ATTR_S	387
31.3.2	ISP_DRC_AUTO_ATTR_S	391
31.3.3	ISP_DRC_ATTR_S	396
<b>32</b>	<b>DIS</b>	<b>401</b>
32.1	功能描述	401
32.2	API 参考	401
32.2.1	CVI_ISP_SetDISAttr	401
32.2.2	CVI_ISP_GetDISAttr	402
32.2.3	CVI_ISP_SetDISConfig	403
32.2.4	CVI_ISP_GetDISConfig	403
32.3	数据类型	404
32.3.1	ISP_DIS_ATTR_S	404
32.3.2	ISP_DIS_CONFIG_S	405
32.3.3	DIS_MODE_E	406
32.3.4	DIS_MOTION_LEVEL_E	407
<b>33</b>	<b>Mono</b>	<b>408</b>
33.1	功能描述	408
33.2	API 参考	408
33.2.1	CVI_ISP_SetMonoAttr	408
33.2.2	CVI_ISP_GetMonoAttr	409
33.3	数据类型	410
33.3.1	ISP_MONO_ATTR_S	410
<b>34</b>	<b>YCONTRAST</b>	<b>411</b>
34.1	功能描述	411
34.2	API 参考	411
34.2.1	CVI_ISP_SetYContrastAttr	411
34.2.2	CVI_ISP_GetYContrastAttr	412

34.3	数据类型	413
34.3.1	ISP_YCONTRAST_MANUAL_ATTR_S	413
34.3.2	ISP_YCONTRAST_AUTO_ATTR_S	414
34.3.3	ISP_YCONTRAST_ATTR_S	414
<b>35</b>	<b>CA</b>	<b>416</b>
35.1	功能描述	416
35.2	API 参考	416
35.2.1	CVI_ISP_SetCAAttr	416
35.2.2	CVI_ISP_GetCAAttr	417
35.3	数据类型	418
35.3.1	ISP_CA_MANUAL_ATTR_S	418
35.3.2	ISP_CA_AUTO_ATTR_S	419
35.3.3	ISP_CA_ATTR_S	419
<b>36</b>	<b>CA2</b>	<b>421</b>
36.1	功能描述	421
36.2	API 参考	421
36.2.1	CVI_ISP_SetCA2Attr	421
36.2.2	CVI_ISP_GetCA2Attr	422
36.3	数据类型	423
36.3.1	ISP_CA2_MANUAL_ATTR_S	423
36.3.2	ISP_CA2_AUTO_ATTR_S	424
36.3.3	ISP_CA2_ATTR_S	424
<b>37</b>	<b>CLUT</b>	<b>426</b>
37.1	功能描述	426
37.2	API 参考	426
37.2.1	CVI_ISP_SetClutAttr	426
37.2.2	CVI_ISP_GetClutAttr	427
37.2.3	CVI_ISP_SetClutSaturationAttr	428
37.2.4	CVI_ISP_GetClutSaturationAttr	428
37.3	数据类型	429
37.3.1	ISP_CLUT_ATTR_S	429
37.3.2	ISP_CLUT_SATURATION_MANUAL_ATTR_S	430
37.3.3	ISP_CLUT_SATURATION_AUTO_ATTR_S	431
37.3.4	ISP_CLUT_SATURATION_ATTR_S	432
<b>38</b>	<b>CSC</b>	<b>433</b>
38.1	功能描述	433
38.2	API 参考	433
38.2.1	CVI_ISP_SetCSCAttr	433
38.2.2	CVI_ISP_GetCSCAttr	434
38.3	数据类型	435
38.3.1	ISP_CSC_ATTR_S	435
38.3.2	ISP_CSC_COLORGAMUT	436
38.3.3	ISP_CSC_MATRX_S	437
<b>39</b>	<b>VC</b>	<b>438</b>
39.1	功能描述	438
39.2	API 参考	438

39.2.1	CVI_ISP_SetVCAAttr	438
39.2.2	CVI_ISP_GetVCAAttr	439
39.3	数据类型	440
39.3.1	ISP_VC_ATTR_S	440
<b>40</b>	<b>统计讯息</b>	<b>441</b>
40.1	概述	441
40.2	API 参考	441
40.2.1	CVI_ISP_SetStatisticsConfig	441
40.2.2	CVI_ISP_GetStatisticsConfig	444
40.2.3	CVI_ISP_GetAESTatistics	444
40.2.4	CVI_ISP_GetWBStatistics	446
40.2.5	CVI_ISP_GetFocusStatistics	446
40.3	数据类型	447
40.3.1	ISP_STATISTICS_CTRL_U	447
40.3.2	ISP_AE_STATISTICS_CFG_S	448
40.3.3	ISP_AE_CROP_S	449
40.3.4	ISP_AE_FACE_CROP_S	450
40.3.5	ISP_WB_STATISTICS_CFG_S	450
40.3.6	ISP_AWB_CROP_S	451
40.3.7	ISP_WB_STATISTICS_S	452
40.3.8	ISP_AWB_GRID_INFO_S	453
40.3.9	ISP_FOCUS_STATISTICS_CFG_S	454
40.3.10	ISP_AF_CFG_S	455
40.3.11	ISP_AF_RAW_CFG_S	456
40.3.12	ISP_AF_PRE_FILTER_CFG_S	456
40.3.13	ISP_AF_CROP_S	457
40.3.14	ISP_AF_H_PARAM_S	458
40.3.15	ISP_AF_V_PARAM_S	458
40.3.16	ISP_STATISTICS_CFG_S	459
40.3.17	ISP_FOCUS_ZONE_S	459
40.3.18	ISP_FE_FOCUS_STATISTICS_S	460
40.3.19	ISP_AF_STATISTICS_S	460
<b>41</b>	<b>查询内部状态消息</b>	<b>462</b>
41.1	概述	462
41.2	API 参考	462
41.2.1	CVI_ISP_QueryInnerStateInfo	462
41.3	数据类型	463
41.3.1	ISP_INNER_STATE_INFO_S	463
<b>42</b>	<b>Debug</b>	<b>465</b>
<b>43</b>	<b>错误码</b>	<b>466</b>
43.1	Proc 调试信息说明	466
43.2	概述	466
43.3	使用方法	466
43.4	ISP	467
43.4.1	LEVEL1 级别调试信息分析	467
43.4.2	LEVEL2 级别调试信息分析	480
43.4.3	LEVEL3 级别调试信息分析	481

<b>44 3A 开发指南概述</b>	<b>483</b>
<b>45 3A 开发用户指南</b>	<b>484</b>
45.1 AF 统计信息使用说明 . . . . .	484
45.1.1 概述 . . . . .	484
45.1.2 输入图像的裁剪 . . . . .	484
45.1.3 Bayer 域的配置 . . . . .	485
45.1.4 抑制光源对于 FV 值的影响 . . . . .	485
45.1.5 统计信息配置注意事项 . . . . .	485
45.1.6 FV 值的获取 . . . . .	485
45.1.7 FV 值的计算 . . . . .	485
45.1.8 FV 计算参考代码 . . . . .	486
<b>46 开发者指南</b>	<b>489</b>
46.1 概述 . . . . .	489
46.2 API 参考 . . . . .	489
46.2.1 isp_sync_task_register . . . . .	489
46.2.2 isp_sync_task_unregister . . . . .	490
46.3 数据类型 . . . . .	491
46.3.1 isp_sync_tsk_method . . . . .	491
46.3.2 isp_sync_task_node . . . . .	492
<b>47 附录</b>	<b>493</b>
<b>48 缩略语</b>	<b>494</b>

## 修订记录

Revision	Date	Description
0.0.1	2020/10/24	初稿
0.0.7	2020/11/24	移除重复的 CCM 章节，增加 AE 范例。
0.0.8	2020/12/24	修改 SatCoringLinearTh 和 SatCoringLinearLmt 的数值范围。
0.0.9	2021/05/05	统一文件格式，新增 182x 相关说明
1.1.1	2021/06/09	修正部分文件说明及描述
1.1.2	2022/01/18	移除不必要的注释
1.2.0	2022/08/18	更新文件格式、相关参数及 API
1.3.0	2022/09/29	修改个别参数注释
1.3.1	2022/10/17	增加对 cv180x 的支持说明
1.3.2	2023/02/06	增加 ISP_SMART_EXPOSURE_ATTR_S 结构体说明
1.3.3	2023/04/20	修改有误文件格式及内容



# 1 声明



## 法律声明

本数据手册包含北京晶视智能科技有限公司（下称“晶视智能”）的保密信息。未经授权，禁止使用或披露本数据手册中包含的信息。如您未经授权披露全部或部分保密信息，导致晶视智能遭受任何损失或损害，您应对因之产生的损失/损害承担责任。

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。本数据手册和本文件所含的所有信息均按“原样”提供，无任何明示、暗示、法定或其他形式的保证。晶视智能特别声明未做任何适销性、非侵权性和特定用途适用性的默示保证，亦对本数据手册所使用、包含或提供的任何第三方的软件不提供任何保证；用户同意仅向该第三方寻求与此相关的任何保证索赔。此外，晶视智能亦不对任何其根据用户规格或符合特定标准或公开讨论而制作的可交付成果承担责任。

## 联系我们

**地址** 北京市海淀区丰豪东路 9 号院中关村集成电路设计园（ICPARK）1 号楼

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

**电话** +86-10-57590723 +86-10-57590724

**邮编** 100094（北京）518100（深圳）

**官方网站** <https://www.sophgo.com/>

**技术论坛** <https://developer.sophgo.com/forum/index.html>

# 2 概述

## 2.1 概述

本文件主要介绍 ISP 的用户接口。可分为系统控制、3A、ISP 模块等三大部分。

第一部分为系统控制，说明如何控制 ISP middleware。

第二部分为 3A，说明如何控制 AE、AWB、AF。本区功能大多可于 Cvi PqTool 中进行调试。

第三部分为 ISP 各个模块，说明如何控制 Black Level Cancellation、Color Correction Matrix、Gamma、Noise Reduction、Sharpeness 等模块。本区功能大多可于 Cvi PqTool 中进行调试。

## 2.2 功能描述

ISP 的控制结构如 图 2.1 所示

1. Lens - 投聚焦光信号，射到 sensor 的感光区域
2. sensor - 经过光电转换，将 Bayer 格式的原始图像送给 ISP
3. ISP - 在此处理 ISP 通过运行在其上的 firmware 对 ISP 逻辑，lens 和 sensor 进行相应控制，进而完成自动光圈、自动曝光、自动白平衡等功能。其中，firmware 的运转靠视频采集单元的中断驱动。PQ Tools 工具通过网口或者串口完成对 ISP 的在线图像质量调节。输出 YUV 域的图像给后端的视频采集单元。

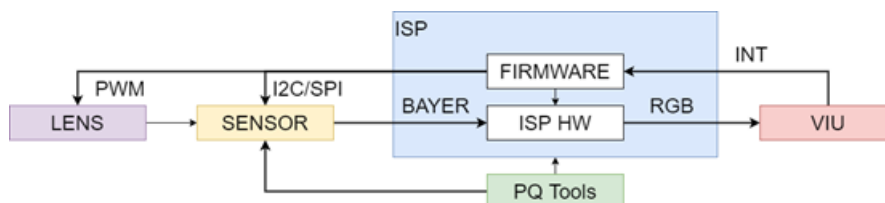


图 2.1: ISP 控制结构示意图

ISP 逻辑主要流程、具体概念和功能点请参见处理器手册。

## 2.3 架构

ISP 的 Firmware 可分为三大部分。

第一部分是 ISP 控制单元和基础算法库。

第二部分是 3A 算法库，目前包括 AE 及 AWB。

第三部分是 sensor 库。

软件架构区分为此三大部分，并且透过注册函数回调以达到可以独立演进的目的。

例如开发者自行实现 3A 函数，只要实作相同的接口，并且注册即可代换 Cvi 预设的 3A 库。

ISP firmware 架构如 图 2.2 所示



图 2.2: ISP firmware 架构

不同的 sensor 都以回调函数的形式，向 ISP 算法库注册控制函数。

ISP 控制单元调度基础算法库和 3A 算法库时，将通过这些回调函数获取初始化参数，并控制 sensor，如调节模拟增益、数字增益、曝光时间。

## 2.4 开发模式

SDK 支持用户使用多种开发模式：

用户使用晶视智能的 3A 算法库。

这时用户需要根据 ISP 基础算法库和 3A 算法库给出的 sensor 适配接口去适配不同的 sensor。

每款 sensor 对应一个文件夹，文件夹中包含两个主要文件：

- sensor\_cmoc.c

该主要实现 ISP 需要的回调函数，包含了 sensor 的适配算法。

不同的 sensor 可能有所不同。

- sensor\_ctrl.c

sensor 的底层控制驱动，主要实现 sensor 的读写和初始化动作。

为了可以同时兼容多个 sensor，所以上两个文件档名会加入 sensor 型号。

例如 Sony imx307 会命名为 imx307\_cmoc.c、imx307\_ctrl.c。

用户可以根据 sensor 的 datasheet 进行这两个文件的开发，必要的时候可以向 sensor 厂家寻求支持。

用户可根据 ISP 库提供的 3A 算法注册接口，实现自己的 3A 算法库开发。

这时用户需要根据 ISP 基础算法库和用户的 3A 算法库给出的 sensor 适配接口去适配不同的 sensor。

用户可以部分使用晶视智能 3A 算法库，部分实现自己的 3A 算法库。

例如 AE 使用晶视智能 AE 算法库 libae.a，AWB 使用自己的 libawb.a 算法库。

## 2.5 内部流程

Firmware 内部流程分两部分，如 图 2.3 所示。

一部分是初始化任务，主要完成 ISP 控制单元的初始化、ISP 基础算法库的初始化、3A 算法库的初始化，包括调用 sensor 的回调获取 sensor 差异化的初始化参数；

另一部分是动态调节过程，在这个过程中，firmware 中的 ISP 控制单元调度 ISP 基础算法库和 3A 算法库，实时计算并进行相应控制。

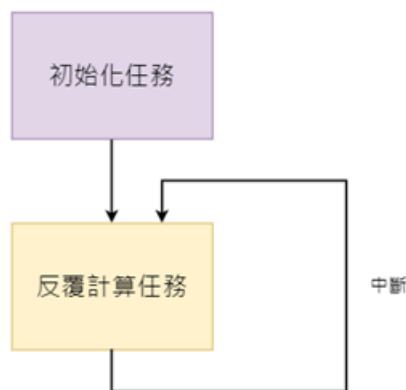


图 2.3: ISP firmware 内部流程

Firmware 的软件结构如 图 2.4 所示。

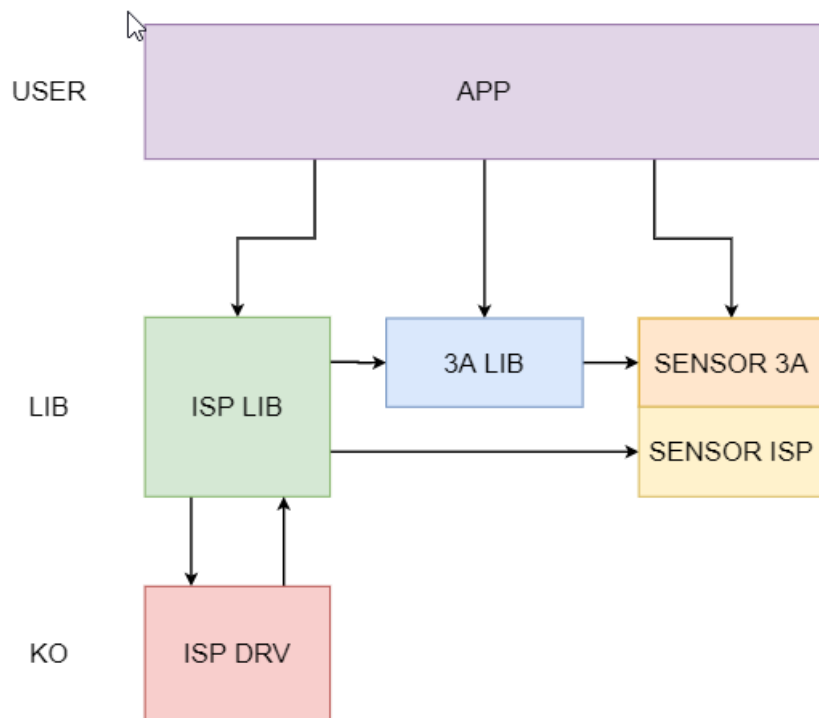


图 2.4: ISP firmware 软件结构

## 2.6 软件流程

软件使用流程如 图 2.5 所示。

PQ Tools 工具主要完成在 PC 端进行动态图像质量调节，可以调节多个影响图像质量的因子，如去噪强度、色彩转换矩阵、饱和度等。

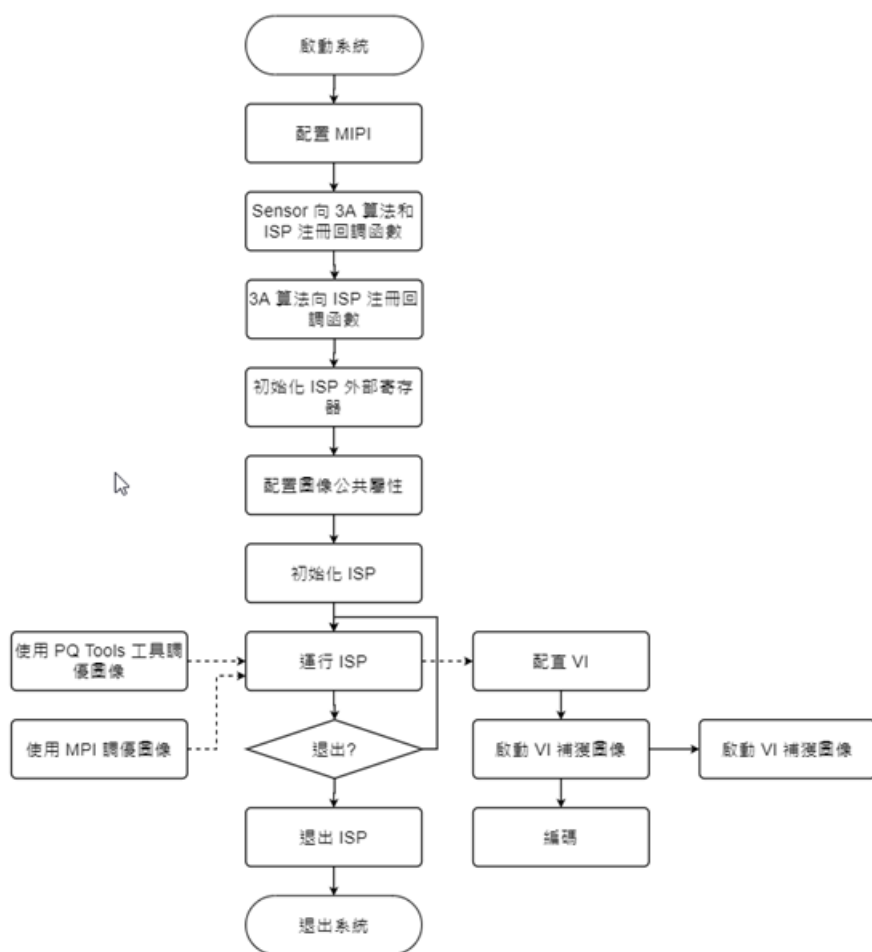


图 2.5: ISP 软件使用流程

如果用户调试好图像效果后，可以使用 PQ Tools 工具提供的配置文件保存功能进行配置参数的保存。

在下次启动时系统可以使用 PQ Tools 工具提供的配置文件加载功能加载已经调节好的图像参数。

# 3 系统控制

## 3.1 功能概述

系统控制部分包含了 ISP 公共属性配置，初始化 ISP Firmware、运行 ISP firmware、退出 ISP firmware，设置 ISP 各模块等功能。

## 3.2 API 参考

本文档中接口，如无特殊说明，不支持多进程。

- `CVI_ISP_MemInit`：初始化 ISP 外部寄存器。
- `CVI_ISP_Init`：初始化 ISP firmware。
- `CVI_ISP_Run`：运行 ISP firmware。
- `CVI_ISP_RunOnce`：运行 ISP firmware 一次。
- `CVI_ISP_Exit`：退出 ISP firmware。
- `CVI_ISP_SetPubAttr`：设置 ISP 公共属性。
- `CVI_ISP_GetPubAttr`：获取 ISP 公共属性。
- `CVI_ISP_SetFMWState`：设置 ISP firmware 状态。
- `CVI_ISP_GetFMWState`：获取 ISP firmware 状态。
- `CVI_ISP_SetModuleControl`：设定 ISP 功能模块的控制。
- `CVI_ISP_GetModuleControl`：获取 ISP 功能模块的控制。
- `CVI_ISP_GetVDTimeOut`：获取 ISP 中断信息。
- `CVI_ISP_SensorRegCallBack`：ISP 提供的 sensor 注册的回调接口。
- `CVI_ISP_SensorUnRegCallBack`：ISP 提供的 sensor 反注册的回调接口。
- `CVI_ISP_AELibRegCallBack`：ISP 提供的 AE 库注册的回调接口。
- `CVI_ISP_AELibUnRegCallBack`：ISP 提供的 AE 库反注册的回调接口。
- `CVI_ISP_AWBLibRegCallBack`：ISP 提供的 AWB 库注册的回调接口。
- `CVI_ISP_AWBLibUnRegCallBack`：ISP 提供的 AWB 库反注册的回调接口。

- `CVI_ISP_SetBindAttr` : 设置 ISP 库与 3A 库、sensor 的绑定关系。
- `CVI_ISP_GetBindAttr` : 获取 ISP 库与 3A 库、sensor 的绑定关系。
- `CVI_ISP_SetCtrlParam` : 设置 ISP 的控制参数。
- `CVI_ISP_GetCtrlParam` : 获取 ISP 的控制参数。
- `CVI_ISP_SetModParam` : 设置 ISP 模块参数。
- `CVI_ISP_GetModParam` : 获取 ISP 模块参数。
- `CVI_BIN_SetBinName` : 设置 PQBIN 存放的路径和文件名
- `CVI_BIN_GetBinName` : 获取 PQBIN 存放的路径和文件名
- `CVI_BIN_GetBinExtraAttr` : 获取 bin 头数据信息。
- `CVI_BIN_GetBinTotalLen` : 获取 bin 数据的总长度。
- `CVI_BIN_ExportBinData` : 将参数存入 PQBin 档。
- `CVI_BIN_ImportBinData` : 从 PQBin 中解析所有模块的数据。
- `CVI_BIN_SaveParamToBin` : 将参数存入 PQBin 档。
- `CVI_BIN_LoadParamFromBin` : 从 PQBin 中解析所有模块的数据。
- `CVI_ISP_IrAutoRunOnce` : 运行红外自动切换功能。
- `CVI_ISP_SetSmartInfo` : 设置智能识别区域信息。
- `CVI_ISP_GetSmartInfo` : 获取智能识别区域信息。

### 3.2.1 CVI\_ISP\_MemInit

#### 【描述】

初始化 ISP 外部寄存器。

#### 【语法】

```
CVI_S32 CVI_ISP_MemInit(VI_PIPE ViPipe);
```

#### 【参数】

参数名称	描述
ViPipe	ViPipe 号

[返回值]

返回值	描述
0	成功。
非 0	失败, 请参见错误码。

#### 【需求】

- 头文件: `cvi_isp.h`



- 库文件: libisp.so

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_Exit](#)

## 3.2.2 CVI\_ISP\_Init

**【描述】**

初始化 ISP firmware。

**【语法】**

```
CVI_S32 CVI_ISP_Init(VI_PIPE ViPipe);
```

**【参数】**

参数名称	描述
ViPipe	ViPipe 号

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: cvi\_isp.h
- 库文件: libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_Exit](#)

### 3.2.3 CVI\_ISP\_Run

**【描述】**

运行 ISP firmware。

**【语法】**

```
CVI_S32 CVI_ISP_Run(VI_PIPE ViPipe);
```

**【参数】**

参数名称	描述
ViPipe	ViPipe 号

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

### 3.2.4 CVI\_ISP\_RunOnce

**【描述】**

运行 ISP firmware 一次。

**【语法】**

```
CVI_S32 CVI_ISP_RunOnce(VI_PIPE ViPipe);
```

**【参数】**

参数名称	描述
ViPipe	ViPipe 号

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

## 3.2.5 CVI\_ISP\_Exit

**【描述】**

退出 ISP firmware。

**【语法】**

```
CVI_S32 CVI_ISP_Exit(VI_PIPE ViPipe);
```

**【参数】**

参数名称	描述
ViPipe	ViPipe 号

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

· [CVI\\_ISP\\_Init](#)

## 3.2.6 CVI\_ISP\_SetPubAttr

**【描述】**

设置 ISP 公共属性。

**【语法】**

```
CVI_S32 CVI_ISP_SetPubAttr(VI_PIPE ViPipe, const ISP_PUB_ATTR_S *pstPubAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstPubAttr	ISP 公共属性。	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

· [CVI\\_ISP\\_GetPubAttr](#)

### 3.2.7 CVI\_ISP\_GetPubAttr

**【描述】**

获取 ISP 公共属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetPubAttr(VI_PIPE ViPipe, ISP_PUB_ATTR_S *pstPubAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstPubAttr	ISP 公共属性。	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetPubAttr](#)

### 3.2.8 CVI\_ISP\_SetFMWState

**【描述】**

设置 ISP firmware 状态。

**【语法】**

```
CVI_S32 CVI_ISP_SetFMWState(VI_PIPE ViPipe, const ISP_FMW_STATE_E enState);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
enState	ISP firmware 状态。	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见错误码。

**【需求】**

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetFMWState](#)

### 3.2.9 CVI\_ISP\_GetFMWState

**【描述】**

获取 ISP firmware 状态。

**【语法】**

```
CVI_S32 CVI_ISP_GetFMWState(VI_PIPE ViPipe, ISP_FMW_STATE_E *penState);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
enState	ISP firmware 状态。	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见错误码。

**【需求】**

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetFMWState](#)

### 3.2.10 CVI\_ISP\_SetModuleControl

**【描述】**

设定 ISP 功能模块的控制。

**【语法】**

```
CVI_S32 CVI_ISP_SetModuleControl(VI_PIPE ViPipe, const ISP_MODULE_CTRL_U_
→*punModCtrl);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
punModCtrl	ISP 功能模块的控制	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetModuleControl](#)

### 3.2.11 CVI\_ISP\_GetModuleControl

**【描述】**

获取 ISP 功能模块的控制。

**【语法】**

```
CVI_S32 CVI_ISP_GetModuleControl(VI_PIPE ViPipe, ISP_MODULE_CTRL_U *punModCtrl);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
punModCtrl	ISP 功能模块的控制。	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetModuleControl](#)

### 3.2.12 CVI\_ISP\_GetVDTimeOut

**【描述】**

获取 ISP 中断信息。

**【语法】**

```
CVI_S32 CVI_ISP_GetVDTimeOut(VI_PIPE ViPipe, ISP_VD_TYPE_E enIspVDType, CVI_
↪ U32 u32MilliSec);
```

**【参数】**



参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
enIspVDType	场同步信号。	输入/输出
u32MilliSec	超时时间, 单位 ms。	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

### 3.2.13 CVI\_ISP\_SensorRegCallBack

**【描述】**

ISP 提供的 sensor 注册的回调接口。

**【语法】**

```
CVI_S32 CVI_ISP_SensorRegCallBack(VI_PIPE ViPipe, ISP_SNS_ATTR_INFO_S_
→ *pstSnsAttrInfo, ISP_SENSOR_REGISTER_S *pstRegister);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstSnsAttrInfo	向 ISP 注册的 Sensor 的属性	输入/输出
pstRegister	Sensor 注册结构体指针	输入/输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

【需求】

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

【注意】

- 此接口在双系统 SDK 的 linux 侧暂不支持。

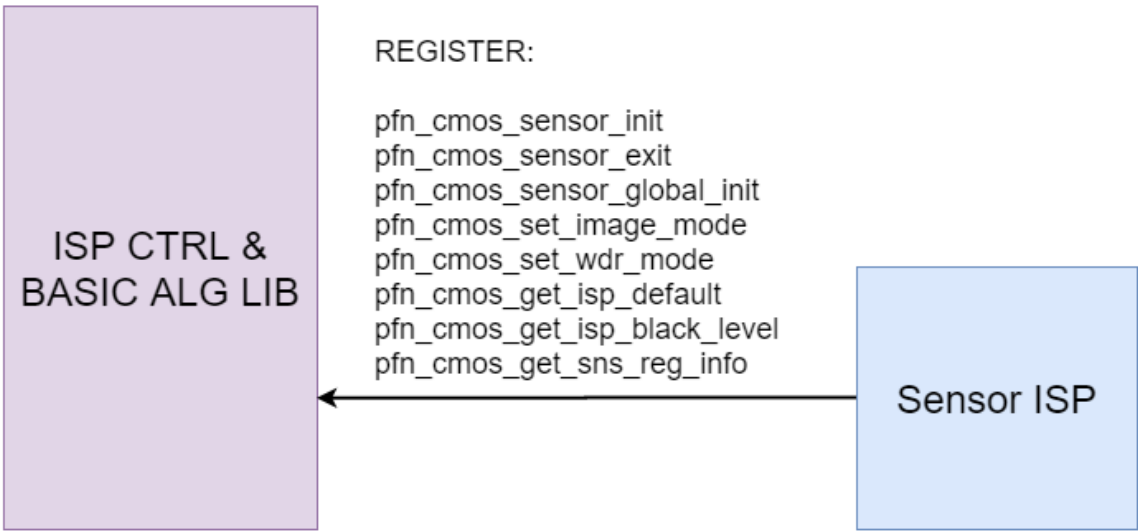


图 3.1: 图 2-1 ISP 库与 sensor 库间的接口

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_SensorUnRegCallback](#)

### 3.2.14 CVI\_ISP\_SensorUnRegCallback

【描述】

ISP 提供的 sensor 反注册的回调接口。

【语法】

```
CVI_S32 CVI_ISP_SensorUnRegCallback(VI_PIPE ViPipe, SENSOR_ID SensorId);
```

【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
SensorId	向 ISP 注册的 Sensor 的 Id。	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SensorRegCallBack](#)

### 3.2.15 CVI\_ISP\_AELibRegCallBack

**【描述】**

ISP 提供的 AE 库注册的回调接口。

**【语法】**

```
CVI_S32 CVI_ISP_AELibRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib, ISP_AE_
→REGISTER_S *pstRegister);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 库结构体指针	输入
pstRegister	AE 库注册结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

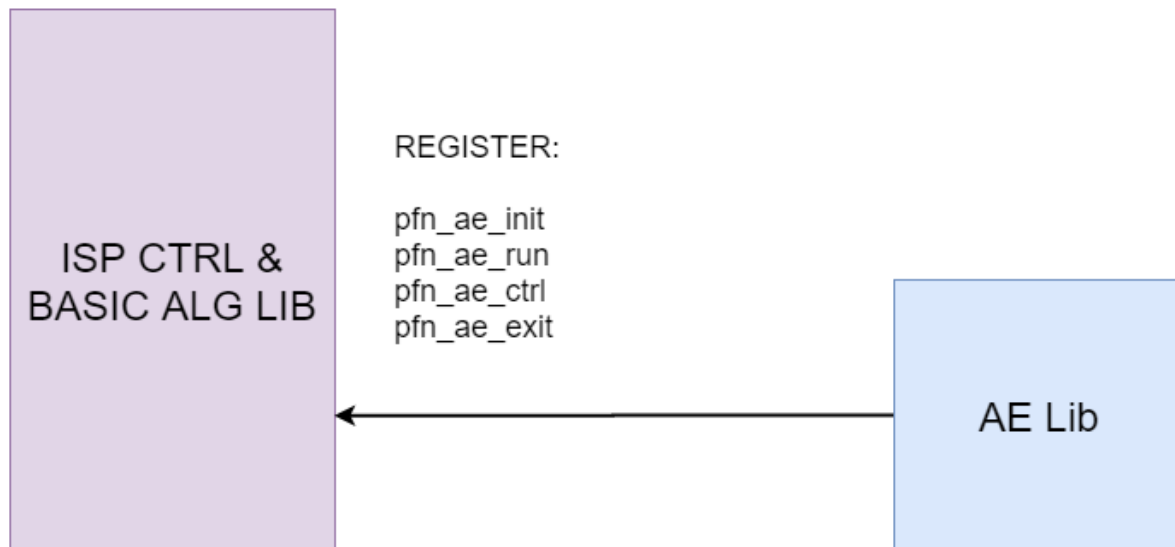


图 3.2: 图 2-2 ISP 库与 AE 库间的接口

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_AELibUnRegCallBack](#)

### 3.2.16 CVI\_ISP\_AELibUnRegCallBack

**【描述】**

ISP 提供的 AE 库反注册的回调接口。

**【语法】**

```
CVI_S32 CVI_ISP_AELibUnRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 库结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_AELibRegCallBack](#)

### 3.2.17 CVI\_ISP\_AWBLibRegCallBack

**【描述】**

ISP 提供的 AWB 库注册的回调接口。

**【语法】**

```
CVI_S32 CVI_ISP_AWBLibRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib, ISP_AWB_
→REGISTER_S *pstRegister);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAwbLib	AWB 库结构体指针	输入
pstRegister	AWB 库注册结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

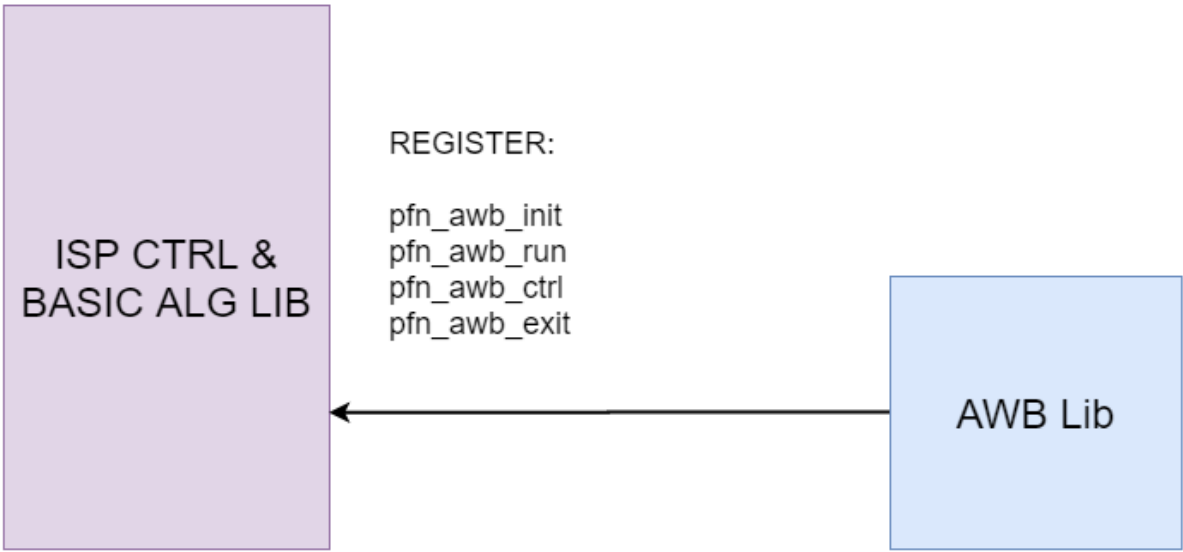


图 3.3: 图 2-3 ISP 库与 AWB 库间的接口

【举例】

无。

【相关主题】

- CVI\_ISP\_AWBLibUnRegCallBack

3.2.18 CVI\_ISP\_AWBLibUnRegCallBack

【描述】

ISP 提供的 AWB 库反注册的回调接口。

【语法】

CVI\_S32 CVI\_ISP\_AWBLibUnRegCallBack(VI\_PIPE ViPipe, ALG\_LIB\_S \*pstAwbLib);

【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAwbLib	AWB 库结构体指针	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件: `cvi_isp.h`
- 库文件: `libisp.a`

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

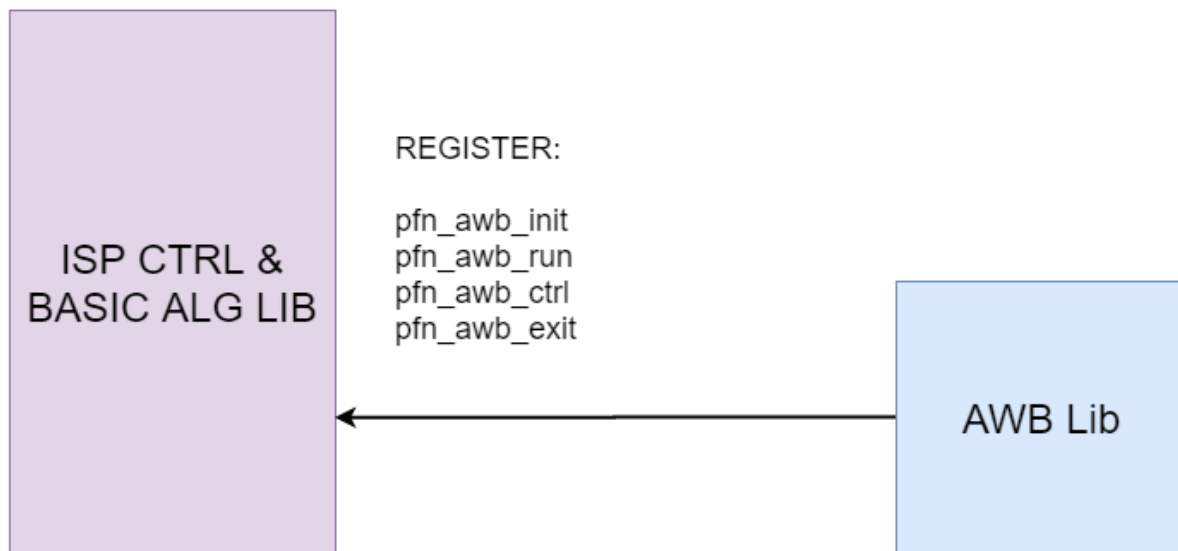


图 3.4: 图 2-3 ISP 库与 AWB 库间的接口

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_AWBLibRegCallBack](#)

### 3.2.19 CVI\_ISP\_SetBindAttr

**【描述】**

设置 ISP 库与 3A 库、sensor 的绑定关系。

**【语法】**

```
CVI_S32 CVI_ISP_SetBindAttr(VI_PIPE ViPipe, const ISP_BIND_ATTR_S *pstBindAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstBindAttr	绑定结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetBindAttr](#)

### 3.2.20 CVI\_ISP\_GetBindAttr

**【描述】**

获取 ISP 库与 3A 库、sensor 的绑定关系。

**【语法】**

```
CVI_S32 CVI_ISP_GetBindAttr(VI_PIPE ViPipe, ISP_BIND_ATTR_S *pstBindAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstBindAttr	绑定结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

- 此接口不支持多进程操作。



- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetBindAttr](#)

### 3.2.21 CVI\_ISP\_SetCtrlParam

**【描述】**

设置 ISP 控制参数。

**【语法】**

```
CVI_S32 CVI_ISP_SetCtrlParam(VI_PIPE ViPipe, const ISP_CTRL_PARAM_S_
↪ *pstIspCtrlParam);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIspCtrlParam	ISP 控制参数结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetCtrlParam](#)

### 3.2.22 CVI\_ISP\_GetCtrlParam

**【描述】**

获取 ISP 控制参数。

**【语法】**

```
CVI_S32 CVI_ISP_GetCtrlParam(VI_PIPE ViPipe, ISP_CTRL_PARAM_S *pstIspCtrlParam);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIspCtrlParam	ISP 控制参数结构体指针。	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetCtrlParam](#)

### 3.2.23 CVI\_ISP\_SetModParam

**【描述】**

设置 ISP 模块参数。

**【语法】**

```
CVI_S32 CVI_ISP_SetModParam(const ISP_MOD_PARAM_S *pstModParam);
```

**【参数】**

参数名称	描述	输入/输出
pstIspModParam	ISP 模块参数结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetModParam](#)

### 3.2.24 CVI\_ISP\_GetModParam

**【描述】**

获取 ISP 模块参数。

**【语法】**

```
CVI_S32 CVI_ISP_GetModParam( ISP_MOD_PARAM_S *pstModParam);
```

**【参数】**

参数名称	描述	输入/输出
pstIspCtrlParam	ISP 模块参数结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_isp.h
- 库文件：libisp.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

· [CVI\\_ISP\\_SetModParam](#)

### 3.2.25 CVI\_BIN\_SetBinName

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.26 CVI\_BIN\_GetBinName

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.27 CVI\_BIN\_GetBinExtraAttr

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.28 CVI\_BIN\_GetBinTotalLen

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.29 CVI\_BIN\_ExportBinData

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.30 CVI\_BIN\_ImportBinData

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.31 CVI\_BIN\_SaveParamToBin

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.32 CVI\_BIN\_LoadParamFromBin

**【说明】**

详情请参见文档《图像质量调试工具使用指南》的 3.1.2 章节。

### 3.2.33 CVI\_ISP\_IrAutoRunOnce

**【描述】**

运行红外自动切换功能。

**【语法】**

```
CVI_S32 CVI_ISP_IrAutoRunOnce(ISP_DEV IspDev, ISP_IR_AUTO_ATTR_S *pstIrAttr);
```

**【参数】**

参数名称	描述	输入/输出
IspDev	ISP 设备号	输入
pstIrAttr	红外自动切换属性	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_comm\_isp.h, cvi\_isp.h
- 库文件：libisp.so

**【注意】**

无。

**【举例】**

```
ISP_DEV IspDev = 0;
ISP_IR_AUTO_ATTR_S stIrAttr;

stIrAttr.bEnable = 1;
stIrAttr.u32Normal2IrIsoThr = 3200;
stIrAttr.u32Ir2NormalIsoThr = 100;
stIrAttr.u32RGMin = 256;
stIrAttr.u32RGMax = 512;
stIrAttr.u32BGMin = 256;
stIrAttr.u32BGMax = 512;
CVI_ISP_IrAutoRunOnce(IspDev, &stIrAttr);
```

**【相关主题】**

无。

### 3.2.34 CVI\_ISP\_SetSmartInfo

**【描述】**

设置 Deep Learning 辨识出的（人脸、人形、物品）坐标给 AE 进行测光。

**【语法】**

```
CVI_S32 CVI_ISP_SetSmartInfo(VI_PIPE ViPipe, const ISP_SMART_INFO_S *pstSmartInfo,
→CVI_U8 TimeOut);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ISP 设备号	输入
pstSmartInfo	AE Deep Learning 识别坐标信息结构指针	输入
TimeOut	识别坐标信息未更新帧数，超过后恢复正常 AE 模式	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

```
//设定人脸在 raw domain坐标位置 (X, Y , W, H) =(0, 0, 100, 100)
//及 frame 的宽/高 1920/1080信息给 AE
//4帧未更新识别结果后恢复正常AE
VI_PIPE ViPipe = 0;
ISP_SMART_INFO_S stSmartInfo;

CVI_ISP_GetSmartInfo(ViPipe, &stSmartInfo);
stSmartInfo.stROI[0].bEnable = 1;
stSmartInfo.stROI[0].bAvailable = 1;
stSmartInfo.stROI[0].u8Num = 1;
stSmartInfo.stROI[0].u16PosX[0] = 0;
stSmartInfo.stROI[0].u16PosY[0] = 0;
stSmartInfo.stROI[0].u16Width[0] = 100;
stSmartInfo.stROI[0].u16Height[0] = 100;
stSmartInfo.stROI[0].u16FrameWidth = 1920;
stSmartInfo.stROI[0].u16FrameHeight = 1080;
CVI_ISP_SetSmartInfo(ViPipe, &stSmartInfo, 4);
```

**【相关主题】**

- [CVI\\_ISP\\_GetSmartInfo](#)

### 3.2.35 CVI\_ISP\_GetSmartInfo

**【描述】**

获取 Deep Learning 辨识出的（人脸、人形、物品）坐标。

**【语法】**

```
CVI_S32 CVI_ISP_GetSmartInfo(VI_PIPE ViPipe, ISP_SMART_INFO_S *pstSmartInfo);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ISP 设备号	输入
pstSmartInfo	AE Deep Learning 识别坐标信息结构指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h

- 库文件: libaec.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetSmartInfo](#)

## 3.3 数据类型

本文档中变量，如未明确指定取值范围，则默认是数据类型对应的取值范围。例如 CVI\_U8 数据类型的变量取值范围为 [0, 255]。本文档中变量，如未明确指定数据精度，则默认是 1。

### 3.3.1 RECT\_S

**【说明】**

定义裁剪窗口起始位置和图像宽高

**【定义】**

```
typedef struct _RECT_S {  
    CVI_S32 s32X;  
    CVI_S32 s32Y;  
    CVI_U32 u32Width;  
    CVI_U32 u32Height;  
} RECT_S;
```

**【成员】**

成员名称	描述
s32X	水平方向起始位置
s32Y	垂直方向起始位置
u32Width	图像宽度
u32Height	图像高度

**【注意事项】**

无。

**【相关数据类型及接口】**

无。



### 3.3.2 SIZE\_S

**【说明】**

定义 sensor 输出的宽高。

**【定义】**

```
typedef struct _SIZE_S {  
    CVI_U32 u32Width;  
    CVI_U32 u32Height;  
} SIZE_S;
```

**【成员】**

成员名称	描述
u32Width	Sensor 输出宽度
u32Height	Sensor 输出高度

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.3 ISP\_BAYER\_FORMAT\_E

**【说明】**

定义了输入 ISP 图像的拜尔阵列的格式类型

**【定义】**

```
typedef enum _ISP_BAYER_FORMAT_E {  
    BAYER_BGGR,  
    BAYER_GBRG,  
    BAYER_GRBG,  
    BAYER_RRGB,  
    //for RGBIR sensor  
    BAYER_GRGBI = 8,  
    BAYER_RGBGI,  
    BAYER_GBGRI,  
    BAYER_BGRGI,  
    BAYER_IGRGB,  
    BAYER_IRGBG,  
    BAYER_IBGRG,  
    BAYER_IGBGR,  
    BAYER_BUTT  
} ISP_BAYER_FORMAT_E;
```

**【成员】**

成员名称	描述
BAYER_XX	各种 bayer 阵列的格式类型, 名称标示了 pixel 以何种方式排列

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.4 WDR\_MODE\_E

**【说明】**

定义了输入 ISP 运行时 sensor 的运行方式,

**【定义】**

```
typedef enum WDR_MODE_E {  
    WDR_MODE_NONE = 0,  
    WDR_MODE_BUILT_IN,  
    WDR_MODE_QUDRA,  
  
    WDR_MODE_2To1_LINE,  
    WDR_MODE_2To1_FRAME,  
    WDR_MODE_2To1_FRAME_FULL_RATE,  
  
    WDR_MODE_3To1_LINE,  
    WDR_MODE_3To1_FRAME,  
    WDR_MODE_3To1_FRAME_FULL_RATE,  
    WDR_MODE_4To1_LINE,  
    WDR_MODE_4To1_FRAME,  
    WDR_MODE_4To1_FRAME_FULL_RATE,  
  
    WDR_MODE_MAX,  
} WDR_MODE_E;
```

**【成员】**

成员名称	描述
WDR_MODE_NONE	线性模式。
WDR_MODE_BUILT_IN	Sensor 合成 WDR 模式。
WDR_MODE_QUDRA	Qudra 模式
WDR_MODE_2To1_LINE	2 帧合成行 WDR 模式。
WDR_MODE_2To1_FRAME	2 帧合成帧 WDR 模式。
WDR_MODE_2To1_FRAME_FULL_RATE	2 帧合成帧 WDR 全帧率模式。
WDR_MODE_3To1_LINE	3 帧合成行 WDR 模式。
WDR_MODE_3To1_FRAME	3 帧合成帧 WDR 模式。
WDR_MODE_3To1_FRAME_FULL_RATE	3 帧合成帧 WDR 全帧率模式。
WDR_MODE_4To1_LINE	4 帧合成行 WDR 模式。
WDR_MODE_4To1_FRAME	4 帧合成帧 WDR 模式。
WDR_MODE_4To1_FRAME_FULL_RATE	4 帧合成帧 WDR 全帧率模式。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.5 ISP\_PUB\_ATTR\_S

**【说明】**

定义 ISP 公共属性

**【定义】**

```
typedef struct _ISP_PUB_ATTR_S {  
    RECT_S stWndRect;  
    SIZE_S stSnsSize;  
    CVI_FLOAT f32FrameRate;  
    ISP_BAYER_FORMAT_E enBayer;  
    WDR_MODE_E enWDRMode;  
    CVI_U8 u8SnsMode;  
} ISP_PUB_ATTR_S;
```

**【成员】**

成员名称	描述
stWndRect	ISP 输出的 rect
stSnsSize	Sensor 的 image size
f32FrameRate	Sensor 的帧率
enBayer	Sensor 的 Bayer 格式
enWDRMode	宽动态模式
u8SnsMode	用于进行 Sensor 初始化序列的选择，在分辨率和帧率相同时，配置不同的 u8SnsMode 对应不同的初始化序列； 其他情况，u8SnsMode 默认配置为 0， 可通过 stSnsSize 和 f32FrameRate 进行初始化序列的选择。

**【注意事项】**

无。

**【相关数据类型及接口】**

- RECT\_S
- SIZE\_S
- ISP\_BAYER\_FORMAT\_E
- WDR\_MODE\_E
- CVI\_ISP\_SetPubAttr
- CVI\_ISP\_GetPubAttr

### 3.3.6 ISP\_FMW\_STATE\_E

**【说明】**

定义 ISP firmware 状态。

**【定义】**

```
typedef enum _ISP_FMW_STATE_E {  
    ISP_FMW_STATE_RUN,  
    ISP_FMW_STATE_FREEZE,  
    ISP_FMW_STATE_BUTT  
} ISP_FMW_STATE_E;
```

**【成员】**

成员名称	描述
ISP_FMW_STATE_RUN	Firmware 正常运行状态
ISP_FMW_STATE_FREEZE	Firmware 冻结状态

**【注意事项】**

无。

【相关数据类型及接口】

- CVI\_ISP\_SetFMWState
- CVI\_ISP\_GetFMWState

### 3.3.7 ISP\_MODULE\_CTRL\_U

【说明】

定义 ISP 功能模块的控制。

【定义】

```
typedef union _ISP_MODULE_CTRL_U {
    CVI_U64 u64Key;
    struct {
        CVI_U64 bitBypassBlc : 1; /* RW:[0] */
        CVI_U64 bitBypassRlsc : 1; /* RW:[1] */
        CVI_U64 bitBypassFpn : 1; /* RW:[2] */
        CVI_U64 bitBypassDpc : 1; /* RW:[3] */
        CVI_U64 bitBypassCrosstalk : 1; /* RW:[4] */
        CVI_U64 bitBypassWBGain : 1; /* RW:[5] */
        CVI_U64 bitBypassDis : 1; /* RW:[6] */
        CVI_U64 bitBypassBnr : 1; /* RW:[7] */
        CVI_U64 bitBypassDemosaic : 1; /* RW:[8] */
        CVI_U64 bitBypassRbgcac : 1; /* RW:[9] */
        CVI_U64 bitBypassMlsc : 1; /* RW:[10] */
        CVI_U64 bitBypassCcm : 1; /* RW:[11] */
        CVI_U64 bitBypassFusion : 1; /* RW:[12] */
        CVI_U64 bitBypassDrc : 1; /* RW:[13] */
        CVI_U64 bitBypassGamma : 1; /* RW:[14] */
        CVI_U64 bitBypassDehaze : 1; /* RW:[15] */
        CVI_U64 bitBypassClut : 1; /* RW:[16] */
        CVI_U64 bitBypassCsc : 1; /* RW:[17] */
        CVI_U64 bitBypassDci : 1; /* RW:[18] */
        CVI_U64 bitBypassCa : 1; /* RW:[19] */
        CVI_U64 bitBypassPreyee : 1; /* RW:[20] */
        CVI_U64 bitBypassMotion : 1; /* RW:[21] */
        CVI_U64 bitBypass3dnr : 1; /* RW:[22] */
        CVI_U64 bitBypassYnr : 1; /* RW:[23] */
        CVI_U64 bitBypassCnr : 1; /* RW:[24] */
        CVI_U64 bitBypassCac : 1; /* RW:[25] */
        CVI_U64 bitBypassCa2 : 1; /* RW:[26] */
        CVI_U64 bitBypassYee : 1; /* RW:[27] */
        CVI_U64 bitBypassYcontrast : 1; /* RW:[28] */
        CVI_U64 bitBypassMono : 1; /* RW:[29] */
        CVI_U64 bitRsv : 34; /* H ; [30:63] */
    };
} ISP_MODULE_CTRL_U;
```

【成员】

成员名称	描述
u64Key	结构体枚举的整形值
bitBypassXXX	各 module 功能控制 bit

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetModuleControl
- CVI\_ISP\_GetModuleControl

### 3.3.8 ISP\_VD\_TYPE\_E

**【说明】**

定义与 ISP 的同步讯号

**【定义】**

```
typedef enum _ISP_VD_TYPE_E {  
    ISP_VD_FE_START = 0,  
    ISP_VD_FE_END,  
    ISP_VD_BE_END,  
    ISP_VD_MAX  
} ISP_VD_TYPE_E;
```

**【成员】**

成员名称	描述
ISP_VD_FE_START	FE 帧的起始讯号
ISP_VD_FE_END	FE 帧的结束讯号
ISP_VD_BE_END	BE 帧的结束讯号

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_GetVDTimeOut

### 3.3.9 ISP\_SNS\_ATTR\_INFO\_S

**【说明】**

定义 ISP sensor 属性。

**【定义】**

```
typedef struct _ISP_SNS_ATTR_INFO_S {  
    CVI_U32 eSensorId;  
} ISP_SNS_ATTR_INFO_S;
```

**【成员】**

成员名称	描述
eSensorId	Sensor ID 号。

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SensorRegCallBack

### 3.3.10 ALG\_LIB\_S

**【说明】**

所用库的信息。

**【定义】**

```
typedef struct _ALG_LIB_S {  
    CVI_S32 s32Id;  
    CVI_CHAR acLibName[ALG_LIB_NAME_SIZE_MAX];  
} ALG_LIB_S;
```

**【成员】**

成员名称	描述
s32Id	算法库实例的 Id。
acLibName	标识算法库名称的字符数组。

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SensorRegCallBack
- CVI\_ISP\_SensorUnRegCallBack
- CVI\_ISP\_AELibRegCallBack

- CVI\_ISP\_AELibUnRegCallBack
- CVI\_ISP\_AWBLibRegCallBack
- CVI\_ISP\_AWBLibUnRegCallBack

### 3.3.11 ISP\_AE\_REGISTER\_S

#### 【说明】

ISP 提供的 AE 库注册的回调接口。

#### 【定义】

```
typedef struct _ISP_AE_REGISTER_S {
    ISP_AE_EXP_FUNC_S stAeExpFunc;
} ISP_AE_REGISTER_S;
```

#### 【成员】

成员名称	描述
stAeExpFunc	AE 注册的回调函数结构体。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- ISP\_AE\_EXP\_FUNC\_S
- CVI\_ISP\_AELibRegCallBack

### 3.3.12 ISP\_AE\_EXP\_FUNC\_S

#### 【说明】

定义 AE 回调函数结构体。

#### 【定义】

```
typedef struct _ISP_AE_EXP_FUNC_S {
    CVI_S32 (*pfn_ae_init)(VI_PIPE ViPipe, const ISP_AE_PARAM_S *pstAeParam);
    CVI_S32 (*pfn_ae_run)(VI_PIPE ViPipe, const ISP_AE_INFO_S *pstAeInfo, ISP_AE_
    →RESULT_S *pstAeResult, CVI_S32 s32Rsv);
    CVI_S32 (*pfn_ae_ctrl)(VI_PIPE ViPipe, CVI_U32 u32Cmd, void *pValue);
    CVI_S32 (*pfn_ae_exit)(VI_PIPE ViPipe);
} ISP_AE_EXP_FUNC_S;
```

#### 【成员】



成员名称	描述
pfn_ae_init	初始化 AE 的回调函数指针。
pfn_ae_run	运行 AE 的回调函数指针。
pfn_ae_ctrl	控制 AE 内部状态的回调函数指针。
pfn_ae_exit	销毁 AE 的回调函数指针。

**【注意事项】**

- 调用 CVI\_ISP\_Init 时将调用 pfn\_ae\_init 回调函数，以初始化 AE 算法库。
- 调用 CVI\_ISP\_Run 时将调用 pfn\_ae\_run 回调函数，以运行 AE 算法库，计算得到 sensor 的曝光时间和增益、ISP 的数字增益。
- 调用 CVI\_ISP\_Exit 时将调用 pfn\_ae\_exit 回调函数，以销毁 AE 算法库。

**【相关数据类型及接口】**

- ISP\_AE\_REGISTER\_S

### 3.3.13 ISP\_AE\_PARAM\_S

**【说明】**

定义 ISP 提供给 AE 的初始化参数结构体。

**【定义】**

```
typedef struct _ISP_AE_PARAM_S{  
    SENSOR_ID SensorId;  
    CVI_U8 u8WDRMode;  
    CVI_U8 u8HDRMode;  
    CVI_U16 u16BlackLevel;  
    CVI_FLOAT f32Fps;  
    ISP_BAYER_FORMAT_E enBayer;  
    ISP_STITCH_ATTR_S stStitchAttr;  
    CVI_S32 s32Rsv;  
    ISP_3AWIN_CONFIG_S aeLEWinConfig[AE_MAX_NUM];  
    ISP_3AWIN_CONFIG_S aeSEWinConfig;  
} ISP_AE_PARAM_S;
```

**【成员】**

成员名称	描述
SensorId	所注册 Sensor 的 id。
u8WDRMode	宽动态模式，ISP 向 AE 提供宽动态模式信息。
u8HDRMode	HDR 模式，ISP 向 AE 提供 HDR 模式信息。
u16BlackLevel	黑电平值，12bit 精度，ISP 向 AE 提供黑电平信息。
f32Fps	帧率，ISP 向 AE 提供帧率信息。
enBayer	Sensor Bayer Pattern，包括 RGGB、GRBG、GBRG、BGGR 四种格式。
stStitchAttr	拼接模式，ISP 向 AE 提供拼接模式信息。
aeLEWinConfig	ISP 向 AE 提供 WDR 长帧的 window 配置信息。
aeSEWinConfig	ISP 向 AE 提供 WDR 短帧的 window 配置信息。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.14 ISP\_SMART\_INFO\_S

**【说明】**

定义 ISP 提供给 AE 的统计信息结构体。

**【定义】**

```
typedef struct _ISP_SMART_INFO_S
{
    ISP_SMART_ROI_S stROI[SMART_CLASS_MAX];
} ISP_SMART_INFO_S;
```

**【成员】**

成员名称	描述
stROI	人脸识别或人形检测区域信息

**【注意事项】**

- 当前仅支持人脸模型与人形模型检测结果，stROI 下标为 0 时为人脸检测结果，下标为 1 时为人形检测结果
- AE 会根据提供的人脸在 raw 的位置来找出人脸位置的 AE window，并使用此 AE window 的亮度来执行 face AE 的测光策略

**【相关数据类型及接口】**

无。

### 3.3.15 ISP\_AE\_INFO\_S

#### 【说明】

定义 ISP 提供给 AE 的统计信息结构体。

#### 【定义】

```
typedef struct _ISP_AE_INFO_S {
    CVI_U32 u32FrameCnt;
    ISP_SMART_ROI_S stSmartInfo;
    ISP_FE_AE_STAT_1_S *pstFEAeStat1[AE_MAX_NUM];
    ISP_FE_AE_STAT_2_S *pstFEAeStat2[AE_MAX_NUM];
    ISP_FE_AE_STAT_3_S *pstFEAeStat3[AE_MAX_NUM];
    ISP_FE_AE_STITCH_STAT_3_S *pstFEAeStiStat;
    ISP_BE_AE_STAT_1_S *pstBEAeStat1;
    ISP_BE_AE_STAT_2_S *pstBEAeStat2;
    ISP_BE_AE_STAT_3_S *pstBEAeStat3;
    ISP_BE_AE_STITCH_STAT_3_S *pstBEAeStiStat;
    ISP_3AWIN_CONFIG_S aeLEWinConfig[AE_MAX_NUM];
    ISP_3AWIN_CONFIG_S aeSEWinConfig;
} ISP_AE_INFO_S;
```

#### 【成员】

成员名称	子成员名称	描述
u32FrameCnt		帧的累加计数
stSmartInfo		请参考 ISP_SMART_INFO_S 接口说明
pstFEAeStat1	u32PixelCount	统计的像素点总个数。
	u32PixelWeight	统计的带权重像素点总个数。
	au32HistogramMemArray	256 段直方图的统计信息数组
pstFEAeStat2	u16GlobalAvgR	全局 R 分量平均值
	u16GlobalAvgGr	全局 Gr 分量平均值
	u16GlobalAvgGb	全局 Gb 分量平均值
	u16GlobalAvgB	全局 B 分量平均值
pstFEAeStat3	au16ZoneAvg	分区间 R、Gr、Gb、B 分量平均值
pstFEAeStiStat	au16ZoneAvg	拼接模式下拼接后分区间 R、Gr、Gb、B 分量平均值
pstBEAeStat1	bStable	目前不使用
pstBEAeStat2	bStable	目前不使用
pstBEAeStat3	bStable	目前不使用
pstBEAeStiStat	bStable	目前不使用
aeLEWinConfig	winWidth	AE window 宽
	winHeight	AE window 高
	winXOffset	AE window 起始位置水平方向偏移值
	winYOffset	AE window 起始位置垂直方向偏移值
	winXNum	AE 水平方向 window 数
	winYNum	AE 垂直方向 window 数

#### 【注意事项】

无。

【相关数据类型及接口】

无。

### 3.3.16 ISP\_AE\_RESULT\_S

【说明】

定义 AE 库返回给 ISP 的配置寄存器结构体。

【定义】

```
typedef struct _ISP_AE_STAT_RESULT_S {  
    CVI_U32 u32IntTime[VI_MAX_PIPE_NUM];  
    CVI_U32 u32IspDgain;  
    CVI_U32 u32Again;  
    CVI_U32 u32Dgain;  
    CVI_U32 u32Iso;  
    CVI_U8 u8AERunInterval;  
    CVI_BOOL bPirisValid;  
    CVI_S32 s32PirisPos;  
    CVI_U32 u32PirisGain;  
    ISP_FSWDR_MODE_E enFSWDRMode;  
    CVI_U32 au32WDRGain[4];  
    CVI_U32 u32HmaxTimes;  
    ISP_AE_STAT_ATTR_S stStatAttr;  
    ISP_DCF_UPDATE_INFO_S stUpdateInfo;  
    CVI_U32 u32ExpRatio;  
    CVI_S16 s16CurrentLV;  
    CVI_U32 u32AvgLuma;  
    CVI_U8 u8MeterFramePeriod;  
    CVI_BOOL bStable;  
    CVI_FLOAT fBvStep;  
    CVI_U32 u32BlcIso;  
    CVI_U32 u32IspDgainSF;  
    CVI_U32 u32AgainSF;  
    CVI_U32 u32DgainSF;  
    CVI_U32 u32IsoSF;  
    CVI_U32 u32BlcIsoSF;  
    CVI_FLOAT fEvRatio[2];  
} ISP_AE_RESULT_S;
```

【成员】

成员名称	描述
u32IntTime	曝光时间
u32IspDgain	ISP 数字增益, 10 bit 精度
u32Again	Sensor 模拟增益, 10 bit 精度
u32Dgain	Sensor 数字增益, 10 bit 精度
u32Iso	AE 总增益值, 2 倍时为 $100 * 2 = 200$
u8AERunInterval	AE 算法运行的间隔
bPirisValid	Piris 是否有效的标志
s32PirisPos	Piris 步进电机的位置, 取值范围与具体 Piris 镜头相关
u32PirisGain	Piris 光圈等效增益, 取值范围与具体 Piris 镜头相关
enFSWDRMode	WDR 合成模式。 0 表示普通多帧合成 WDR 模式; 1 表示长帧模式; 2 表示自动长帧模式。
au32WDRGain	兼容参数, 目前不使用
u32HmaxTimes	Sensor 对应读出一行的时间, 单位: ns
stStatAttr	兼容参数, 目前不使用
stUpdateInfo	用于传递 AE 相关 DCF 信息
u32ExpRatio	WDR 模式时, 长/短帧的曝光比值, 64 为 1 倍
s16CurrentLV	AE 目前估算的环境亮度, 数值越大表示环境越亮
u32AvgLuma	目前画面的亮度
u8MeterFramePeriod	AE 的曝光生效周期
bStable	当前 AE 收敛状态是否稳定
fBvStep	目前画面的亮度距离目标亮度的步幅
u32BlcIso	长曝对应 BLC 参考的总增益值, 此增益仅包含 sensor 的仿真/数字增益, 不包含 ISP 数字增益
u32IspDgainSF	WDR 模式, 短帧 ISP 数字增益, 10 bit 精度
u32AgainSF	WDR 模式, 短帧 Sensor 模拟增益, 10 bit 精度
u32DgainSF	WDR 模式, 短帧 Sensor 数字增益, 10 bit 精度
u32IsoSF	WDR 模式, AE 总增益值
u32BlcIsoSF	短曝 BLC 参考的总增益值, 此增益仅包含 sensor 的仿真/数字增益, 不包含 ISP 数字增益
fEvRatio	AE 当前帧与前一帧的曝光差异比率

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.17 ISP\_SENSOR\_EXP\_FUNC\_S

#### 【说明】

定义 sensor 回调函数结构体。

#### 【定义】

```
typedef struct ISP_SENSOR_EXP_FUNC_S {
    CVI_VOID (*pfn_cmos_sensor_init)(VI_PIPE ViPipe);
    CVI_VOID (*pfn_cmos_sensor_exit)(VI_PIPE ViPipe);
    CVI_VOID (*pfn_cmos_sensor_global_init)(VI_PIPE ViPipe);
    CVI_S32 (*pfn_cmos_set_image_mode)(VI_PIPE ViPipe, ISP_CMOS_SENSOR_IMAGE_
→MODE_S *pstSensorImageMode);
    CVI_S32 (*pfn_cmos_set_wdr_mode)(VI_PIPE ViPipe, CVI_U8 u8Mode);

    /* the algs get data which is associated with sensor, except 3a */
    CVI_S32 (*pfn_cmos_get_isp_default)(VI_PIPE ViPipe, ISP_CMOS_DEFAULT_S *pstDef);
    CVI_S32 (*pfn_cmos_get_isp_black_level)(VI_PIPE ViPipe, ISP_CMOS_BLACK_LEVEL_S_
→*pstBlackLevel);
    CVI_S32 (*pfn_cmos_get_sns_reg_info)(VI_PIPE ViPipe, ISP_SNS_SYNC_INFO_S_
→*pstSnsRegsInfo);

    /* the function of sensor set pixel detect */
    //CVI_VOID (*pfn_cmos_set_pixel_detect)(VI_PIPE ViPipe, bool bEnable);
} ISP_SENSOR_EXP_FUNC_S;
```

#### 【成员】

成员名称	描述
pfn_cmos_sensor_init	初始化 sensor 的回调函数指针。
pfn_cmos_sensor_exit	sensor 的回调退出函数指针。
pfn_cmos_sensor_global_init	初始化全局变量的回调函数指针。
pfn_cmos_set_image_mode	设置分辨率和帧率切换的回调函数指针。
pfn_cmos_set_wdr_mode	设置 wdr 模式的回调函数指针。
pfn_cmos_get_isp_default	获取 ISP 基础算法的初始值的回调函数指针。
pfn_cmos_get_isp_black_level	获取 sensor 的黑电平值的回调函数指针，支持根据 sensor 增益动态调整黑电平值
pfn_cmos_get_sns_reg_info	获取 sensor 寄存器信息的回调函数指针，用于实现内核态配置 AE 信息
pfn_cmos_set_pixel_detect	设置坏点校正开关的回调函数指针

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### 3.3.18 ISP\_SENSOR\_REGISTER\_S

**【说明】**

ISP 提供的 sensor 注册的回调接口。

**【定义】**

```
typedef struct bmISP_SENSOR_REGISTER_S {  
    ISP_SENSOR_EXP_FUNC_S stSnsExp;  
} ISP_SENSOR_REGISTER_S;
```

**【成员】**

成员名称	描述
stSnsExp	Sensor 注册的回调函数结构体。

**【注意事项】**

无。

**【相关数据类型及接口】**

- ISP\_SENSOR\_EXP\_FUNC\_S
- CVI\_ISP\_SensorRegCallBack

### 3.3.19 ISP\_AWB\_EXP\_FUNC\_S

**【说明】**

定义 AWB 回调函数结构体。

**【定义】**

```
typedef struct ISP_AWB_EXP_FUNC_S {  
    CVI_S32 (*pfn_awb_init)(VI_PIPE ViPipe, const ISP_AWB_PARAM_S *pstAwbParam);  
    CVI_S32 (*pfn_awb_run)(VI_PIPE ViPipe, const ISP_AWB_INFO_S *pstAwbInfo, ISP_AWB_  
→RESULT_S *pstAwbResult, CVI_S32 s32Rsv);  
    CVI_S32 (*pfn_awb_ctrl)(VI_PIPE ViPipe, CVI_U32 u32Cmd, CVI_VOID * pValue);  
    CVI_S32 (*pfn_awb_exit)(VI_PIPE ViPipe);  
} ISP_AWB_EXP_FUNC_S;
```

**【成员】**

成员名称	描述
pfn_awb_init	初始化 AWB 的回调函数指针
pfn_awb_run	运行 AWB 的回调函数指针。
pfn_awb_ctrl	控制 AWB 内部状态的回调函数指针。
pfn_awb_exit	销毁 AWB 的回调函数指针

**【注意事项】**

无。

【相关数据类型及接口】

无。

### 3.3.20 ISP\_AWB\_REGISTER\_S

【说明】

定义 AWB 注册结构体。

【定义】

```
typedef struct _ISP_AWB_REGISTER_S {  
    ISP_AWB_EXP_FUNC_S stAwbExpFunc;  
} ISP_AWB_REGISTER_S;
```

【成员】

成员名称	描述
stAwbExpFunc	AWB 注册回调函数的结构体

【注意事项】

无。

【相关数据类型及接口】

无。

### 3.3.21 ISP\_AWB\_PARAM\_S

【说明】

定义 ISP 给 AWB 初始化参数结构

【定义】

```
typedef struct _ISP_AWB_PARAM_S {  
    SENSOR_ID SensorId;  
    CVI_U8 u8WDRMode;  
    CVI_U8 u8AWBZoneRow;  
    CVI_U8 u8AWBZoneCol;  
    CVI_U8 u8AWBZoneBin;  
    ISP_STITCH_ATTR_S stStitchAttr;  
    CVI_U16 u16AWBWidth;  
    CVI_U16 u16AWBHeight;  
    CVI_S8 s8Rsv;  
} ISP_AWB_PARAM_S;
```

【成员】



成员名称	描述
SensorId	向 ISP 注册的 Sensor ID
u8WDRMode	宽动态模式
u8AWBZoneRow	AWB 统计行数
u8AWBZoneCol	AWB 统计列数
u8AWBZoneBin	AWB 统计亮度数
stStitchAttr	拼接讯息结构体
u16AWBWidth	AWB 算法的图像宽度
u16AWBHeight	AWB 算法的图像高度
s8Rsv	保留参数

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.22 ISP\_AWB\_STAT\_1\_S

**【说明】**

定义 AWB 统计值讯息结构体

**【定义】**

```
typedef struct _ISP_AWB_STAT_1_S {  
    CVI_U16 u16MeteringAwbAvgR;  
    CVI_U16 u16MeteringAwbAvgG;  
    CVI_U16 u16MeteringAwbAvgB;  
    CVI_U16 u16MeteringAwbCountAll;  
} ISP_AWB_STAT_1_S;
```

**【成员】**

成员名称	描述
u16MeteringAwbAvgR	Bayer 域全局统计中白点的 R 分量
u16MeteringAwbAvgG	Bayer 域全局统计中白点的 G 分量
u16MeteringAwbAvgB	Bayer 域全局统计中白点的 B 分量
u16MeteringAwbCountAll	Bayer 域全局统计中白点的个数

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.23 ISP\_AWB\_STAT\_RESULT\_S

**【说明】**

定义 AWB 统计值讯息结构体

**【定义】**

```
typedef struct _ISP_AWB_STAT_RESULT_S {  
    CVI_U16 *pau16ZoneAvgR;  
    CVI_U16 *pau16ZoneAvgG;  
    CVI_U16 *pau16ZoneAvgB;  
    CVI_U16 *pau16ZoneCount;  
} ISP_AWB_STAT_RESULT_S;
```

**【成员】**

成员名称	描述
pau16ZoneAvgR	Bayer 域分区统计值中白点 R 分量数组的起始位置
pau16ZoneAvgG	Bayer 域分区统计值中白点 G 分量数组的起始位置
pau16ZoneAvgB	Bayer 域分区统计值中白点 B 分量数组的起始位置
pau16ZoneCount	Bayer 域分区统计值中白点个数数组的起始位置

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.24 ISP\_AWB\_INFO\_S

**【说明】**

定义 ISP 提供给 AWB 统计值讯息的结构体

**【定义】**

```
typedef struct _ISP_AWB_INFO_S {  
    CVI_U32 u32FrameCnt;  
    ISP_SMART_ROI_S stSmartInfo;  
    ISP_AWB_STAT_1_S *pstAwbStat1[ISP_CHANNEL_MAX_NUM];  
    ISP_AWB_STAT_RESULT_S stAwbStat2[ISP_CHANNEL_MAX_NUM];  
    CVI_U8 u8AwbGainSwitch;  
    CVI_U32 au32WDRWBGain[ISP_BAYER_CHN_NUM];  
    CVI_U32 u32IsoNum;  
    CVI_S16 s16LVx100;
```

(下页继续)

(续上页)

```
CVI_FLOAT fBVstep;
}ISP_AWB_INFO_S;
```

**【成员】**

成员名称	描述
u32FrameCnt	帧的累加计数
stSmartInfo	特殊区域的信息 (给车牌或是人脸使用), 请参考 ISP_SMART_INFO_S 接口说明
pstAwbStat1	AWB 统计讯息 1
stAwbStat2	AWB 统计讯息 2
u8AwbGainSwitch	保留, 无作用
au32WDRWBGain	保留, 无作用
u32IsoNum	当前画面的 ISO 增益
s16LVx100	当前画面的亮度值 x100
fBVstep	当前画面 AE 收敛的差值

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.25 ISP\_SMART\_ROI\_S

**【说明】**

定义 ISP 提供给 AE 人脸识别或人形识别区域统计信息的结构体

**【定义】**

```
typedef struct _ISP_SMART_ROI_S {
    CVI_BOOL bEnable;
    CVI_BOOL bAvailable;
    CVI_U8  u8Luma;
    CVI_U8  u8Num;
    CVI_U16 u16PosX[SMART_MAX_NUM];
    CVI_U16 u16PosY[SMART_MAX_NUM];
    CVI_U16 u16Width[SMART_MAX_NUM];
    CVI_U16 u16Height[SMART_MAX_NUM];
    CVI_U16 u16FrameWidth;
    CVI_U16 u16FrameHeight;
} ISP_SMART_ROI_S;
```

**【宏】**

```
# define SMART_MAX_NUM (3)
```

**【成员】**

成员名称	描述
bEnable	模型是否使能
bAvailable	模型是否有可用的检测结果
u8Luma	兼容参数, 目前不使用
u16PosX[SMART_MAX_NUM]	检测出的人脸位置对应应在 raw 上的 X 坐标
u16PosY[SMART_MAX_NUM]	检测出的人脸位置对应应在 raw 上的 Y 坐标
u16Width[SMART_MAX_NUM]	检测出的人脸位置对应应在 raw 上的宽
u16Height[SMART_MAX_NUM]	检测出的人脸位置对应应在 raw 上的高
u16FrameWidth	Raw 的 frame 宽
u16FrameHeight	Raw 的 frame 高

**【注意事项】**

- 当前仅支持人脸模型与人形模型检测结果, stROI 下标为 0 时为人脸检测结果, 下标为 1 时为人形检测结果
- AE 会根据提供的人脸在 raw 的位置来找出人脸位置的 AE window, 并使用此 AE window 的亮度来执行 face AE 的测光策略

**【相关数据类型及接口】**

无。

### 3.3.26 ISP\_AWB\_RAW\_STAT\_ATTR\_S

**【说明】**

定义 AWB lib 返回给 ISP 的配置缓存器结构体

**【定义】**

```
typedef struct ISP_AWB_RAW_STAT_ATTR_S {
    CVI_BOOL bStatCfgUpdate;
    CVI_U16 u16MeteringWhiteLevelAwb;
    CVI_U16 u16MeteringBlackLevelAwb;
    CVI_U16 u16MeteringCrRefMaxAwb;
    CVI_U16 u16MeteringCbRefMaxAwb;
    CVI_U16 u16MeteringCrRefMinAwb;
    CVI_U16 u16MeteringCbRefMinAwb;
} ISP_AWB_RAW_STAT_ATTR_S;
```

**【成员】**

成员名称	描述
bStatCfgUpdate	保留, 无作用
u16MeteringWhiteLevelAwb	保留, 无作用
u16MeteringBlackLevelAwb	保留, 无作用
u16MeteringCrRefMaxAwb	保留, 无作用
u16MeteringCbRefMaxAwb	保留, 无作用
u16MeteringCrRefMinAwb	保留, 无作用
u16MeteringCbRefMinAwb	保留, 无作用

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.27 ISP\_AWB\_RESULT\_S

**【说明】**

定义 AWB lib 返回给 ISP 配置缓存器的结构体

**【定义】**

```
typedef struct _ISP_AWB_RESULT_S {  
    CVI_U32 au32WhiteBalanceGain[ISP_BAYER_CHN_NUM];  
    CVI_U16 au16ColorMatrix[CCM_MATRIX_SIZE];  
    CVI_U32 u32ColorTemp;  
    CVI_U8 u8Saturation[4];  
    ISP_AWB_RAW_STAT_ATTR_S stRawStatAttr;  
    CVI_BOOL bStable;  
} ISP_AWB_RESULT_S;
```

**【成员】**

成员名称	描述
au32WhiteBalanceGain	AWB lib 算出来的 R,Gr,Gb,B 颜色通道的增益
au16ColorMatrix	保留, 无作用
u32ColorTemp	现在 AWB 估算出的色温
u8Saturation	当前饱和度
stRawStatAttr	保留, 无作用
bStable	是否收敛

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 3.3.28 ISP\_BIND\_ATTR\_S

**【说明】**

定义 ISP 库与 Sensor、3A 库之间绑定关系的结构体。

**【定义】**

```
typedef struct _ISP_BIND_ATTR_S {  
    CVI_S32 sensorId;  
    ALG_LIB_S stAeLib;  
    ALG_LIB_S stAfLib;  
    ALG_LIB_S stAwbLib;  
} ISP_BIND_ATTR_S;
```

**【成员】**

成员名称	描述
SensorId	注册 Sensor 的 Id。
stAeLib	AE 库结构体。
stAwbLib	AF 库结构体。
stAfLib	AWB 库结构体。

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetBindAttr
- CVI\_ISP\_GetBindAttr

### 3.3.29 ISP\_CTRL\_PARAM\_S

**【说明】**

定义 ISP 控制参数结构体。

**【定义】**

```
typedef struct _ISP_CTRL_PARAM_S {  
    CVI_U32 u32AESTatIntvl;  
    CVI_U32 u32AWBStatIntvl;  
    CVI_U32 u32AFStatIntvl;  
    CVI_U32 u32ProcParam;  
    CVI_U32 u32ProcLevel;  
    CVI_U32 u32UpdatePos;  
    CVI_U32 u32IntTimeOut;  
    CVI_U32 u32PwmNumber;  
    CVI_U32 u32PortIntDelay;  
} ISP_CTRL_PARAM_S;
```

**【成员】**

成员名称	描述
u32AStatIntvl	ISP 3A AE 统计信息更新频率，单位为帧 取值范围：(0,0xffffffff]
u32AWBStatIntvl	ISP 3A AWB 统计信息更新频率，单位为帧 取值范围：(0,0xffffffff]
u32AFStatIntvl	ISP 3A AF 统计信息更新频率，单位为帧 取值范围：(0,0xffffffff]
u32ProcParam	ISP 收集信息的收集频率，单位为帧，默认值为 30，这个值越高，ISP 的 RISC-V 占用率就越低 取值范围：(0,0xffffffff]
u32ProcLevel	ISP 的 proc 打印 Level，Level 为 0,proc 功能关闭，level 为 1，精简信息，level 为 2，较多信息，level 为 3，非常多信息（会把 3A 统计值也打印出来）
u32UpdatePos	目前 cvitek 仅支持在帧开始配置 sensor 寄存器，默认值为 0
u32IntTimeOut	表示中断超时的时间（ms）。目前 cvitek 没有使用
u32PwmNumber	表示 PWM 号。目前 cvitek 没有使用
u32PortIntDelay	表示 Port 中断延时时间

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCtrlParam
- CVI\_ISP\_GetCtrlParam

### 3.3.30 ISP\_MOD\_PARAM\_S

**【说明】**

定义 ISP 模块参数结构体。

**【定义】**

```
typedef struct _ISP_MOD_PARAM_S {  
    CVI_U32 u32IntBotHalf;  
} ISP_MOD_PARAM_S;
```

**【成员】**

成员名称	描述
u32IntBotHalf	表示 ISP 中断处理是否采用下半部机制，默认值为 0, 目前 cvitek 仅支持（读统计信息和配置 sensor 和 ISP 同步寄存器）在中断服务程序中完成；

**【注意事项】**

无。

**【相关数据类型及接口】**

- [CVI\\_ISP\\_SetModParam](#)
- [CVI\\_ISP\\_GetModParam](#)

### 3.3.31 ISP\_IR\_AUTO\_ATTR\_S

**【说明】**

定义红外自动切换属性。

**【定义】**

```
typedef struct _ISP_IR_AUTO_ATTR_S
{
    CVI_BOOL bEnable;
    CVI_U32 u32Normal2IrIsoThr;
    CVI_U32 u32Ir2NormalIsoThr;
    CVI_U32 u32RGMax;
    CVI_U32 u32RGMin;
    CVI_U32 u32BGMax;
    CVI_U32 u32BGMin;
    ISP_IR_STATUS_E enIrStatus;
    ISP_IR_SWITCH_STATUS_E enIrSwitch;
} ISP_IR_AUTO_ATTR_S;
```

**【成员】**



成员名称	描述
bEnable	红外自动切换使能。 CVI_FALSE: 关闭 CVI_TRUE: 使能。
u32Normal2IrIsoThr	从普通状态切换到红外状态的 ISO 阈值。当实际生效的 ISO 大于此阈值时，系统需要切换到红外状态。 取值范围: [0, 0xFFFFFFFF]。
u32Ir2NormalIsoThr	从红外状态切换到普通状态的 ISO 阈值。当实际生效的 ISO 小于此阈值时，系统需要切换到普通状态。 取值范围: [0, 0xFFFFFFFF]。
u32RGMax	红外状态下的 R/G 最大值。实际图像的 R/G 大于此参数时，系统需要切换到普通状态。4.8 格式。 取值范围: [0, 0xFFF]。
u32RGMin	红外状态下的 R/G 最小值。实际图像的 R/G 小于此参数时，系统需要切换到普通状态。4.8 格式。 取值范围: [0, u32RGMax]。
u32BGMax	红外状态下的 B/G 最大值。实际图像的 B/G 大于此参数时，系统需要切换到普通状态。4.8 格式。 取值范围: [0, 0xFFF]。
u32BGMin	红外状态下的 B/G 最小值。实际图像的 B/G 小于此参数时，系统需要切换到普通状态。4.8 格式。 取值范围: [0, u32BGMax]。
enIrStatus	设备当前的红外状态。应配置为设备实际的红外状态，需要用户保证状态的正确性。
enIrSwitch	设备的红外切换状态，为只读。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

# 4 AE

## 4.1 概述

CVI AE 模块实现的功能是：根据自动测光系统获得当前图像的曝光量，再自动配置镜头光圈、sensor 快门及增益来获得最佳的图像质量。

自动曝光的算法主要分光圈优先、快门优先、增益优先。

光圈优先时算法会优先调整光圈到合适的位置，再分配曝光时间和增益，只适合 p-iris 镜头，这样能均衡噪声和景深。

快门优先时算法会优先分配曝光时间，再分配 sensor 增益和 ISP 增益，这样拍摄的图像噪声会比较小。

增益优先则是优先分配 sensor 增益和 ISP 增益，再分配曝光时间，适合拍摄运动物体的场景。

当前 AE 算法也支持客户设定更灵活的曝光分配策略，AE 模块的工作流程如图 4.1 所示

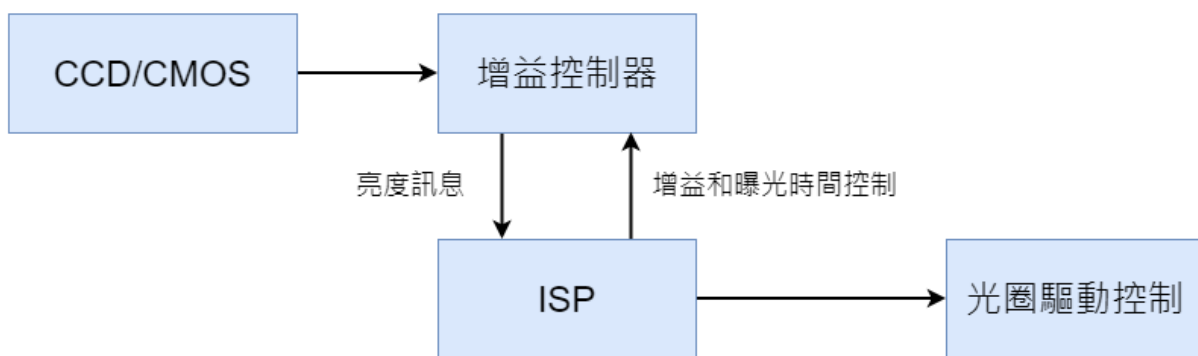


图 4.1: AE 模块工作流程图

## 4.2 重要概念

- 曝光时间：sensor 积累电荷的时间，是 sensor pixel 从开始曝光到电量被读出的这段时间。
- 曝光增益：对 sensor 输出电荷的总放大系数，一般有数字增益和模拟增益，模拟增益引入的噪声会稍小，所以一般优先用模拟增益。
- 光圈：光圈是镜头中可以改变中间孔大小的机械装置。
- 抗闪烁：由于电灯电源工频与 sensor 的帧率不匹配而导致的画面闪烁，一般通过限定曝光时间和修改 sensor 的帧率来达到抗闪烁的效果。

## 4.3 功能描述

AE 模块主要有 ISP 的 AE 统计信息模块及 AE 控制策略的 AE 算法 Firmware 两部分组成。

ISP 的 AE 统计信息模块主要是提供 sensor 输入数据的亮度信息统计。

其提供的统计信息包括直方图和平均值，可同时提供整幅图像的 256 段的直方图和 R/Gr/Gb/B 四分量平均值统计信息，

还可提供将整幅图像分成 MxN 区块，每个区块的 R/Gr/Gb/B 四分量平均值统计信息，具体如图 4.2 所示

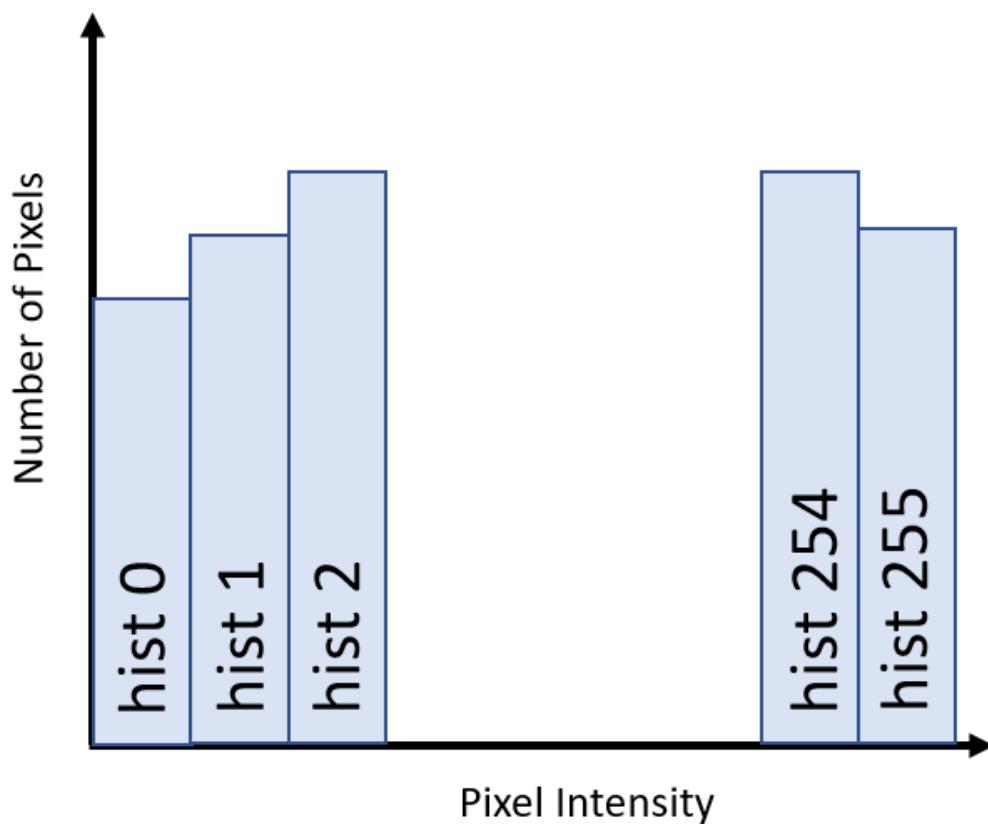


图 4.2: AE 256 段统计信息直方图

AE 算法的主要工作原理是即时获取输入图像的统计信息并与设定目标亮度进行比较，从而动态调节 sensor 的曝光时间和增益以及镜头光圈大小以达到实际亮度与设定目标亮度接近。其工作原理如图 4.3 所示。

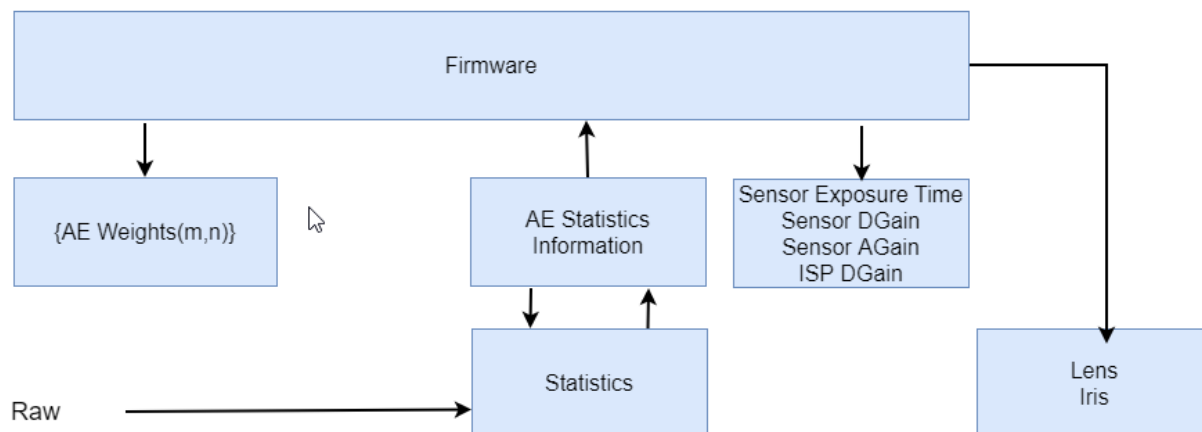


图 4.3: AE 工作原理图

## 4.4 API 参考

### 4.4.1 AE 库接口

所有 AE 库接口都只是针对 CVI AE 库，如果客户自己实现 AE 库，不需要关注这些接口，且无法使用这些接口。

- `CVI_AE_Register`：向 ISP 注册 AE 库。
- `CVI_AE_UnRegister`：向 ISP 反注册 AE 库。
- `CVI_AE_SensorRegCallBack`：AE 库提供的 sensor 注册的回调接口。
- `CVI_AE_SensorUnRegCallBack`：AE 库提供的 sensor 反注册的回调接口。

#### 4.4.1.1 CVI\_AE\_Register

##### 【描述】

向 ISP 注册 AE 库。

##### 【语法】

```
CVI_S32 CVI_AE_Register(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib);
```

##### 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 算法库结构体指针	输入

##### 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

##### 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

##### 【注意】

- 此接口在双系统 SDK 的 linux 侧暂不支持。

##### 【举例】

无。

##### 【相关主题】

- `CVI_AE_UnRegister`

#### 4.4.1.2 CVI\_AE\_UnRegister

**【描述】**

向 ISP 反注册 AE 库。

**【语法】**

```
CVI_S32 CVI_AE_UnRegister(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 算法库结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

- 该接口调用了 ISP 库提供的 AE 反注册回调接口，以实现 AE 向 ISP 库反注册的功能。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- CVI\_AE\_Register

#### 4.4.1.3 CVI\_AE\_SensorRegCallBack

**【描述】**

AE 库提供的 sensor 注册的回调接口。

**【语法】**

```
CVI_S32 CVI_AE_SensorRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib, ISP_SNS_
↪ATTR_INFO_S *pstSnsAttrInfo, AE_SENSOR_REGISTER_S *pstRegister);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 算法库结构体指针	输入
pstSnsAttrInfo	向 AE 注册的 Sensor 的属性	输入
pstRegister	Sensor 注册结构体指针	输入

## 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

## 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

## 【注意】

- SensorId 是 sensor 库中自定义的值，主要用于校对向 ISP 注册的 sensor 和向 3A 注册的 sensor 是否为同一个 sensor。
- AE 通过 sensor 注册的一系列回调接口，获取差异化的初始化参数，并控制 sensor。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

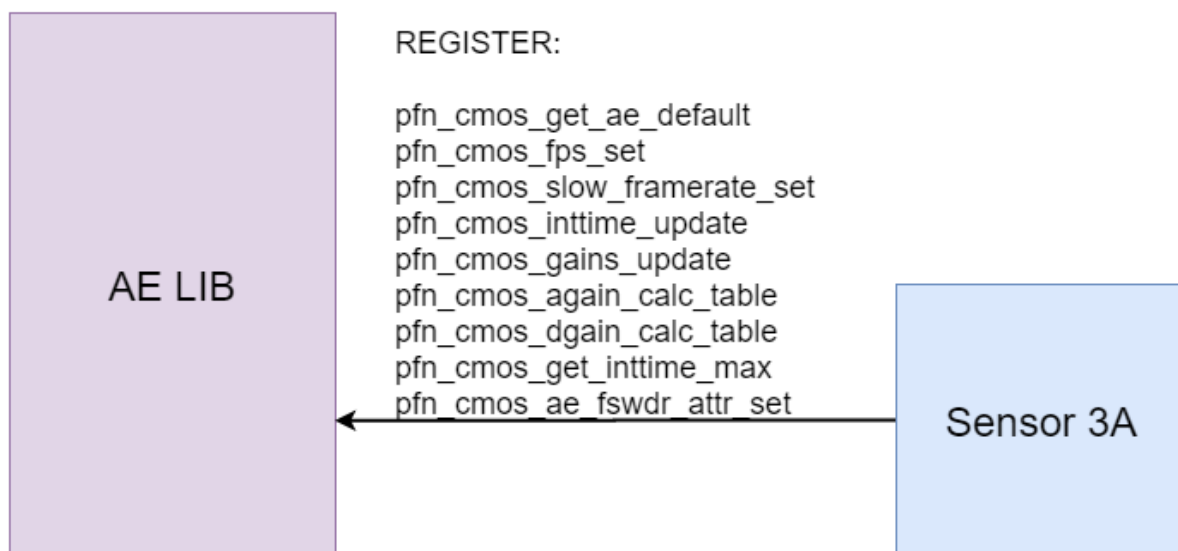


图 4.4: 图 3-4 AE 库与 sensor 库间的接口

## 【举例】

无。

**【相关主题】**

- CVI\_AE\_SensorUnRegCallBack

**4.4.1.4 CVI\_AE\_SensorUnRegCallBack****【描述】**

AE 库提供的 sensor 反注册的回调接口。

**【语法】**

```
CVI_S32 CVI_AE_SensorUnRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib, SENSOR_ID_↵  
↵SensorId);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AE 算法库结构体指针	输入
SensorId	向 AE 反注册的 Sensor 的 Id	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

- SensorId 是 sensor 库中自定义的值，主要用于校对向 ISP 注册的 sensor 和向 3A 注册的 sensor 是否为同一个 sensor。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

- CVI\_AE\_SensorRegCallBack



## 4.4.2 AE 控制模块

曝光控制接口：

- CVI\_ISP\_SetExposureAttr：设置 AE 曝光属性。
- CVI\_ISP\_GetExposureAttr：获取 AE 曝光属性。
- CVI\_ISP\_SetWDRExposureAttr：设置 WDR 模式下的 AE 曝光属性。
- CVI\_ISP\_GetWDRExposureAttr：获取 WDR 模式下的 AE 曝光属性。
- CVI\_ISP\_SetAERouteAttr：设置 AE 路由属性。
- CVI\_ISP\_GetAERouteAttr：获取 AE 路由属性。
- CVI\_ISP\_SetAERouteAttrEx：设置 AE 曝光分配扩展属性，支持分别设置 AE 分配策略中的 sensor 模拟增益，sensor 数字增益和 ISP 数字增益。
- CVI\_ISP\_GetAERouteAttrEx：获取 AE 曝光分配策略扩展属性。
- CVI\_ISP\_QueryExposureInfo：获取 AE 内部状态信息。
- CVI\_ISP\_SetAntiFlicker：设置 AE anti flicker 功能。
- CVI\_ISP\_GetAntiFlicker：获取 AE anti flicker 设定。
- CVI\_ISP\_QueryFps：获取 AE 当前的 fps。
- CVI\_ISP\_GetCurrentLvX100：获取当前环境亮度的 LV 值。
- CVI\_ISP\_SetSmartExposureAttr：设置智能模式下的 AE 曝光属性。
- CVI\_ISP\_GetSmartExposureAttr：获取智能模式下的 AE 曝光属性。
- CVI\_ISP\_SetAERouteSFAttr：设置 AE WDR 短帧曝光分配策略。
- CVI\_ISP\_GetAERouteSFAttr：获取 AE WDR 短帧曝光分配策略。
- CVI\_ISP\_SetAERouteSFAttrEx：设置 AE WDR 短帧曝光分配扩展属性。
- CVI\_ISP\_GetAERouteSFAttrEx：获取 AE WDR 短帧曝光分配扩展属性。

### 4.4.2.1 CVI\_ISP\_SetExposureAttr

#### 【描述】

设定 AE 曝光属性。

#### 【语法】

```
CVI_S32 CVI_ISP_SetExposureAttr(VI_PIPE ViPipe, const ISP_EXPOSURE_ATTR_S_
↪ *pstExpAttr)
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstExpAttr	AE 曝光属性结构体指针	输入

## 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

## 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

## 【注意】

- AE 曝光控制类型为自动时，曝光时间，曝光增益都由 AE 算法自动控制，可以通过配置自动曝光属性结构体 stAuto 里面的参数得到不同的曝光效果。
- AE 曝光控制类型为手动时，可以通过配置手动曝光属性结构体 stManual 控制使能类型（曝光时间使能、ISO num 使能、sensor 模拟增益使能、sensor 数字增益使能、ISP 数字增益使能）及相应的曝光参数（曝光时间、ISO Num、sensor 模拟增益、sensor 数字增益、ISP 数字增益）。
- AE 曝光控制类型为自动时，配置手动曝光属性的参数无效。同理，AE 曝光控制类型为手动时，配置自动曝光属性的参数无效。
- AE 曝光控制类型为手动时，若曝光参数设置超出最大（小）值，将使用 sensor 支持的最大（小）值代替。
- 无论是自动曝光还是手动曝光，曝光时间的单位为微秒 (us)，曝光增益的单位为 10bit 精度的倍数，即 1024 代表 1 倍，2048 代表 2 倍等。

## 【举例】

```
// 自动曝光属性设置
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Max = 40000;
stExpAttr.stAuto.stExpTimeRange.u32Min = 10;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

stExpAttr.stAuto.u8Speed = 0x40;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

stExpAttr.stAuto.enAESTrategyMode = AE_EXP_HIGHLIGHT_PRIOR;
stExpAttr.stAuto.u16HistRatioSlope = 0x8;
stExpAttr.stAuto.u8MaxHistOffset = 0x10;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

stExpAttr.stAuto.stAntiflicker.bEnable = CVI_TRUE;
stExpAttr.stAuto.stAntiflicker.enFrequency = AE_FREQUENCY_50HZ;
stExpAttr.stAuto.stAntiflicker.enMode = ISP_ANTIFLICKER_NORMAL_MODE;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);
```

(下页继续)

(续上页)

```

stExpAttr.stAuto.stAEDelayAttr.u16BlackDelayFrame = 10;
stExpAttr.stAuto.stAEDelayAttr.u16WhiteDelayFrame = 0;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//手动曝光属性设置(使用 gain 控制):
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enAGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enDGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISPDGainOpType = OP_TYPE_MANUAL;

stExpAttr.stManual.enGainType = AE_TYPE_GAIN;
stExpAttr.stManual.u32AGain = 0x400;
stExpAttr.stManual.u32DGain = 0x400;
stExpAttr.stManual.u32ISPDGain = 0x400;
stExpAttr.stManual.u32ExpTime = 0x40000;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//手动曝光属性设置(使用 ISO Num 控制):
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISONumOpType = OP_TYPE_MANUAL;

stExpAttr.stManual.enGainType = AE_TYPE_ISO;
stExpAttr.stManual.u32ISONum = 1600;
stExpAttr.stManual.u32ExpTime = 0x40000;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//自动模式设定最大增益 32x(使用 gain 控制)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.enGainType = AE_TYPE_GAIN;
stExpAttr.stAuto.stSysGainRange.u32Max = 32767;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//自动模式设定最大增益 32x(使用 ISO Num 控制)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.enGainType = AE_TYPE_ISO;
stExpAttr.stAuto.stISONumRange.u32Max = 3200;

```

(下页继续)

(续上页)

```
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//快门模式(33333 us)设定最大增益 32x
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Min = 33333;
stExpAttr.stAuto.stExpTimeRange.u32Max = 33333;
stExpAttr.stAuto.enGainType = AE_TYPE_ISO;
stExpAttr.stAuto.stISONumRange.u32Max = 3200;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//手动模式设固定增益(32x, 使用 ISO Num)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enGainType = AE_TYPE_ISO;
stExpAttr.stManual.enISONumOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.u32ISONum = 3200;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//手动模式设固定增益(32x, 使用 Gain)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enGainType = AE_TYPE_ISO;
stExpAttr.stManual.enAGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enDGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISPDGainOpType = OP_TYPE_MANUAL;

stExpAttr.stManual.u32AGain = 0x8000;
stExpAttr.stManual.u32DGain = 0x400;
stExpAttr.stManual.u32ISPDGain = 0x400;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//自动快门
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//手动设定快门 (16384 us)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
```

(下页继续)

(续上页)

```
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.u32ExpTime = 16384;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//快门模式 (16384 us)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Min = 16384;
stExpAttr.stAuto.stExpTimeRange.u32Max = 16384;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//设定 antiflicker 50Hz
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.stAuto.stAntiflicker.bEnable = 1;
stExpAttr.stAuto.stAntiflicker.enMode = ISP_ANTIFLICKER_NORMAL_MODE;
stExpAttr.stAuto.stAntiflicker.enFrequency = AE_FREQUENCY_50HZ;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//设定各 LV 的目标亮度
#define TABLE_SIZE 21
CVI_U8 target_max[TABLE_SIZE] = {15, 15, 15, 15,
    15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60, 60,
    60, 60};
CVI_U8 target_min[TABLE_SIZE] = {40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45,
    50, 50, 50, 50, 50, 50, 60, 60, 60};

CVI_U8 i;
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);

for (i = 0; i < TABLE_SIZE; i++) {
    stExpAttr.stAuto.au8AdjustTargetMin[i] = target_min[i];
    stExpAttr.stAuto.au8AdjustTargetMax[i] = target_max[i];
}
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);
```

**【相关主题】**

- CVI\_ISP\_GetExposureAttr

#### 4.4.2.2 CVI\_ISP\_GetExposureAttr

**【描述】**

获取 AE 曝光属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetExposureAttr(VI_PIPE ViPipe, ISP_EXPOSURE_ATTR_S *pstExpAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstExpAttr	AE 曝光属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetExposureAttr

#### 4.4.2.3 CVI\_ISP\_SetWDRExposureAttr

**【描述】**

设置 WDR 模式下的 AE 曝光属性。

**【语法】**

```
CVI_S32 CVI_ISP_SetWDRExposureAttr(VI_PIPE ViPipe, const ISP_WDR_EXPOSURE_
→ATTR_S *pstWDRExpAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstWDRExpAttr	WDR 模式下的 AE 曝光属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

```
//手动设定曝光比 8x
VI_PIPE ViPipe = 0;

ISP_WDR_EXPOSURE_ATTR_S stWdrExpAttr;
CVI_ISP_GetWDRExposureAttr(ViPipe, &stWdrExpAttr);
stWdrExpAttr.enExpRatioType = OP_TYPE_MANUAL;
stWdrExpAttr.au32ExpRatio[0] = 512;
CVI_ISP_SetWDRExposureAttr(ViPipe, &stWdrExpAttr);

//设定自动模式曝光比限制在 8x ~ 16x
ISP_WDR_EXPOSURE_ATTR_S stWdrExpAttr;
CVI_ISP_GetWDRExposureAttr(ViPipe, &stWdrExpAttr);
stWdrExpAttr.enExpRatioType = OP_TYPE_AUTO;
stWdrExpAttr.u32ExpRatioMin = 512;
stWdrExpAttr.u32ExpRatioMax = 1024;
CVI_ISP_SetWDRExposureAttr(ViPipe, &stWdrExpAttr);

//设定长/短帧各 LV 的目标亮度
#define TABLE_SIZE 21
CVI_U8 LeTarget[TABLE_SIZE] = {15, 15, 15, 15,
    15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60, 60,
    60, 60};

CVI_U8 SeTarget[TABLE_SIZE] = {5, 5, 5, 5,
    5, 5, 5, 10, 10, 15, 15, 15, 15, 20, 20, 20, 20, 20, 20,
    20, 20};

CVI_U8 i;
ISP_WDR_EXPOSURE_ATTR_S stWdrExpAttr;
CVI_ISP_GetWDRExposureAttr(ViPipe, &stWdrExpAttr);

for (i = 0; i < TABLE_SIZE; i++) {
```

(下页继续)

(续上页)

```
stWdrExpAttr.au8LEAdjustTargetMin[i] =  
stWdrExpAttr.au8LEAdjustTargetMax[i] = LeTarget[i];  
  
stWdrExpAttr.au8SEAdjustTargetMin[i] =  
stWdrExpAttr.au8SEAdjustTargetMax[i] = SeTarget[i];  
}  
CVI_ISP_SetWDRExposureAttr(ViPipe, &stWdrExpAttr);
```

**【相关主题】**

- [CVI\\_ISP\\_GetWDRExposureAttr](#)

**4.4.2.4 CVI\_ISP\_GetWDRExposureAttr****【描述】**

获取 WDR 模式下的 AE 曝光属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetWDRExposureAttr(VI_PIPE ViPipe, ISP_WDR_EXPOSURE_ATTR_S_  
→*pstWDRExpAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstWDRExpAttr	WDR 模式下的 AE 曝光属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetWDRExposureAttr](#)



#### 4.4.2.5 CVI\_ISP\_SetAERouteAttr

**【描述】**

设置 AE 曝光分配策略。

**【语法】**

```
CVI_S32 CVI_ISP_SetAERouteAttr(VI_PIPE ViPipe, const ISP_AE_ROUTE_S_  
→*pstAERouteAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteAttr	AE 曝光分配策略结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

此接口用于设定 AE 曝光分配路线，AE 计算得到的曝光量将按照设定的路线进行分配，用户可以根据自己的需求设定曝光优先、增益优先、光圈优先。

AE 分配路线遵循以下规则：

- 最大支持 16 个节点，每个节点有曝光时间、增益、光圈三个分量，增益包含模拟增益、数字增益、ISP 数字增益。
- 节点中曝光时间的单位为 us，不能设置为 0，也不能设置太小导致实际对应的曝光行数为 0，否则可能产生异常。
- 光圈分量仅支持 P-Iris，不支持 DC-Iris，因为 DC-Iris 无法精确控制，所以 DC-Iris 和手动光圈镜头光圈分量是无效的。即光圈类型为 DC-Iris 时，节点光圈分量不会对曝光量分配产生任何影响。
- 节点的曝光量是曝光时间、增益和光圈的乘积，节点曝光量需为单调递增，后一个节点的曝光量应大于或等于前一个节点的曝光量，第一个节点的曝光量最小，最后一个节点的曝光量最大。
- 如果相邻节点的曝光量增加，那么应该有一个分量增加，其他分量固定，增加的分量决定该段路线的分配策略。例如曝光时间分量增加，那么该段路线的分配策略是曝光时间优先。
- 不支持设置等曝光量节点。

用户可以根据不同的场景设置不同的路线，分配路线支持动态切换。

- 针对 DC-Iris 和手动光圈镜头，默认 AE 分配策略是首先分配曝光时间，其次分配增益。针对 P-Iris 镜头，默认 AE 分配策略是首先调节光圈，将光圈调至最大后调节曝光时间，最后再分配增益。如果当前曝光量不在用户设定的路线范围当中，按默认策略分配。
- 在线进行 DC-Iris 和 P-Iris 切换，AE route 会重置为与光圈类型相匹配的默认分配策略，用户可以根据需要在切换光圈类型时自行设置 AE route。
- 自动降帧时，最大曝光时间的改变会更新到分配路线中。
- 帧率切换时，若用户设置的最大曝光目标时间大于切换后 1 帧所允许的最大曝光时间，那么分配路线的最大曝光时间会更新为切换后 1 帧所允许的最大曝光时间。
- 发生自动降帧、线性与 WDR 模式切换、帧率切换、限制曝光时间或增益的最大最小值等情况时，实际生效的 AE route 可能与 MPI 设置的不一致，此时可以通过 CVI\_ISP\_QueryExposureInfo 获取实际生效的 AE route。

**【举例】**

```
//设定 gain 128x 的 route
VI_PIPE ViPipe = 0;

ISP_AE_ROUTE_S stRoute;
CVI_ISP_SetAERouteAttr(ViPipe, &stRoute);

stRoute.u32TotalNum = 3;

stRoute.astRouteNode[0].u32IntTime = 30;
stRoute.astRouteNode[0].u32SysGain = 1024;
stRoute.astRouteNode[0].enIrisFNO = 10;
stRoute.astRouteNode[0].u32IrisFNOLin = 1024;

stRoute.astRouteNode[1].u32IntTime = 33333;
stRoute.astRouteNode[1].u32SysGain = 1024;
stRoute.astRouteNode[1].enIrisFNO = 10;
stRoute.astRouteNode[1].u32IrisFNOLin = 1024;

stRoute.astRouteNode[2].u32IntTime = 33333;
stRoute.astRouteNode[2].u32SysGain = 131072;
stRoute.astRouteNode[2].enIrisFNO = 10;
stRoute.astRouteNode[2].u32IrisFNOLin = 1024;

CVI_ISP_SetAERouteAttr(ViPipe, &stRoute);
```

**【相关主题】**

- [CVI\\_ISP\\_GetAERouteAttr](#)
- [ISP\\_AE\\_ROUTE\\_S](#)

#### 4.4.2.6 CVI\_ISP\_GetAERouteAttr

**【描述】**

获取 AE 曝光分配策略。

**【语法】**

```
CVI_S32 CVI_ISP_GetAERouteAttr(VI_PIPE ViPipe, ISP_AE_ROUTE_S *pstAERouteAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteAttr	AE 曝光分配策略结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetAERouteAttr

#### 4.4.2.7 CVI\_ISP\_SetAERouteAttrEx

**【描述】**

设置 AE 曝光分配扩展属性，支持分别设置 AE 分配策略中的 sensor 模拟增益，sensor 数字增益和 ISP 数字增益。

**【语法】**

```
CVI_S32 CVI_ISP_SetAERouteAttrEx(VI_PIPE ViPipe, const ISP_AE_ROUTE_EX_S_  
↪ *pstAERouteAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteAttrEx	AE 曝光分配策略扩展属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件: `cvi_ae.h`
- 库文件: `libae.a`

**【注意】**

- 此接口用于设定 AE 曝光分配扩展属性，AE 计算得到的曝光量将按照设定的路线进行分配，用户可以根据自己的需求设定曝光时间优先、sensor 模拟增益优先、sensor 数字增益优先、ISP 数字增益优先和光圈优先。该接口可用于设置 WDR 模式下的曝光分配路线，减轻正常室内照度多帧合成 WDR 产生的工频闪现象，优化 WDR 模式图像效果。
- AE 曝光分配扩展属性是否生效可通过配置 `CVI_ISP_SetExposureAttr` 接口中的 `bAERouteExValid` 来实现。`bAERouteExValid` 为 `CVI_TRUE` 时使用扩展 AE route，否则使用正常 AE route。
- 其它注意事项与 `CVI_ISP_SetAERouteAttr` 的规则相同

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetAERouteAttrEx](#)
- [ISP\\_AE\\_ROUTE\\_EX\\_S](#)

**4.4.2.8 CVI\_ISP\_GetAERouteAttrEx****【描述】**

获取 AE 曝光分配策略扩展属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetAERouteAttrEx(VI_PIPE ViPipe, ISP_AE_ROUTE_EX_S_
→ *pstAERouteAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteAttrEx	AE 曝光分配策略扩展属性结构体指针	输出

## 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

## 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

## 【注意】

无。

## 【举例】

```
VI_PIPE ViPipe = 0;

ISP_EXPOSURE_ATTR_S stExpAttr;
ISP_AE_ROUTE_EX_S stRouteEx;
CVI_U32 au32RouteExNode[6][5]
= {{ 30, 1024, 1024, 1024, 0},
{ 30, 1024, 1024, 1024, 10},
{ 30, 16384, 1024, 1024, 10},
{400000, 16384, 1024, 1024, 10},
{400000, 16384, 4096, 1024, 10},
{400000, 16384, 4096, 4096, 10}};
CVI_ISP_GetAERouteAttrEx(ViPipe, &stRouteEx);
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bAERouteExValid = CVI_TRUE;
stRouteEx.u32TotalNum = 6;
memcpy(stRouteEx.astRouteExNode, au32RouteExNode, sizeof(au32RouteExNode));
```

## 【相关主题】

- [CVI\\_ISP\\_SetAERouteAttrEx](#)

## 4.4.2.9 CVI\_ISP\_QueryExposureInfo

## 【描述】

获取 AE 内部状态信息，包括 256 段直方图和平均亮度等统计信息，同时还可获取 AE 运行状态中的曝光时间、增益、曝光量、当帧的亮度、目前的环境亮度和实际生效的 AE route 等信息。

## 【语法】

```
CVI_S32 CVI_ISP_QueryExposureInfo(VI_PIPE ViPipe, ISP_EXP_INFO_S *pstExpInfo);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstExpInfo	曝光内部状态信息结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: `cvi_ae.h`
- 库文件: `libae.a`

**【注意】**

获取的曝光时间以微秒 (us) 为单位，获取的 sensor 模拟增益、sensor 数字增益和 ISP 数字增益以倍数为单位，精度是 10bit。

**【举例】**

```
// get sensor again, sensor dgain, isp dgain and exposure ratio
VI_PIPE ViPipe = 0;
ISP_EXP_INFO_S stExpInfo;

CVI_ISP_QueryExposureInfo(ViPipe, &stExpInfo);

printf("sensor used time = %d\n", stExpInfo.u32ExpTime);
printf("sensor used again = %d\n", stExpInfo.u32AGain);
printf("sensor used dgain = %d\n", stExpInfo.u32DGain);
printf("isp used dgain = %d\n", stExpInfo.u32ISPDGain);
printf("WDR Exposure ratio = %d\n", stExpInfo.u32WDRExpRatio);
printf("Light Value = %f\n", stExpInfo.fLightValue);
```

**【相关主题】**

无。

**4.4.2.10 CVI\_ISP\_SetAntiFlicker****【描述】**

设置 AE anti flicker 功能。

**【语法】**

```
CVI_S32 CVI_ISP_SetAntiFlicker(VI_PIPE ViPipe, CVI_BOOL enable, CVI_U8 frequency);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
enable	AE anti flicker 使能	输入
frequency	抗闪频率	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

```
//设定 AE anti flicker 50Hz
VI_PIPE ViPipe = 0;
CVI_U8 enable, frequency;
enable = 1;
frequency = 50;

CVI_ISP_SetAntiFlicker(ViPipe, enable, frequency);
```

**【相关主题】**

- CVI\_ISP\_GetAntiFlicker

**4.4.2.11 CVI\_ISP\_GetAntiFlicker****【描述】**

获取 AE anti flicker 设定。

**【语法】**

```
CVI_S32 CVI_ISP_GetAntiFlicker(VI_PIPE ViPipe, CVI_BOOL *pEnable, CVI_U8 *pFrequency);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pEnable	AE anti flicker 使能指针	输入
pFrequency	抗闪频率指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件: `cvi_ae.h`
- 库文件: `libae.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

#### 4.4.2.12 CVI\_ISP\_QueryFps

**【描述】**

获取 AE 当前 fps。

**【语法】**

```
CVI_S32 CVI_ISP_QueryFps(VI_PIPE ViPipe, CVI_FLOAT *pFps);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pFps	AE fps 指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见错误码。

**【需求】**

- 头文件: `cvi_ae.h`
- 库文件: `libae.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_QueryExposureInfo](#)



## 4.4.2.13 CVI\_ISP\_GetCurrentLvX100

## 【描述】

用于获取当前 AE 的测光结果，即环境光亮度值 LV。可用于判断黑夜和白天，达到类似光敏电阻的效果。

## 【语法】

```
CVI_S32 CVI_ISP_GetCurrentLvX100(VI_PIPE ViPipe, CVI_S16 *ps16Lv);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
ps16Lv	返回当前 LV 的值	输出

## 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

## 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

## 【注意】

无。

## 【举例】

```
#include "cvi_ae.h"

#define ENTER_NIGHT_LV_LEVEL 0
#define ENTER_DAY_LV_LEVEL 700
#define CHECK_COUNT 5

bool checkDayOrNight(void)
{
    static bool dayOrNight = true;
    static CVI_U8 checkDayCount = 0;
    static CVI_U8 checkNightCount = 0;
    CVI_S16 lv = 0;

    CVI_ISP_GetCurrentLvX100(0, &lv);
    if (lv > ENTER_DAY_LV_LEVEL) {
        if (checkDayCount < CHECK_COUNT) {
            checkDayCount++;
        } else {
            dayOrNight = true;
        }
    }
}
```

(下页继续)

(续上页)

```
    checkNightCount = 0;
} else if (lv < ENTER_NIGHT_LV_LEVEL) {
    if (checkNightCount < CHECK_COUNT) {
        checkNightCount++;
    } else {
        dayOrNight = false;
    }
    checkDayCount = 0;
} else {
    checkDayCount = 0;
    checkNightCount = 0;
}
return dayOrNight;
}
int main(void)
{
    while (1) {
        sleep(1);
        printf("checkDayOrNight: %s\n", checkDayOrNight() ? "day" : "night");
    }
    return 0;
}
```

**【相关主题】**

无。

**4.4.2.14 CVI\_ISP\_SetSmartExposureAttr****【描述】**

设置智能模式下的 AE 曝光属性。仅在有智能信息时生效。

**【语法】**

```
CVI_S32 CVI_ISP_SetSmartExposureAttr(VI_PIPE ViPipe, const ISP_SMART_EXPOSURE_
→ATTR_S *pstSmartExpAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstSmartExpAttr	智能模式下的 AE 曝光属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件: cvi\_ae.h
- 库文件: libae.a

**【注意】**

无。

**【举例】**

```
#include "cvi_ae.h"

VI_PIPE ViPipe = 0;
ISP_SMART_EXPOSURE_ATTR_S stSmartExpAttr;

CVI_ISP_GetSmartExposureAttr(ViPipe, &stSmartExpAttr);

stSmartExpAttr.bEnable = 1;
stSmartExpAttr.u8LumaTarget = 56;
stSmartExpAttr.u16ExpCoef = 1024;
stSmartExpAttr.u16ExpCoefMax = 4096;
stSmartExpAttr.u16ExpCoefMin = 256;
stSmartExpAttr.u8SmartInterval = 1;
stSmartExpAttr.u8SmartSpeed = 32;
stSmartExpAttr.u16SmartDelayNum = 5;
stSmartExpAttr.u8Weight = 80;
stSmartExpAttr.u8NarrowRatio = 75;

CVI_ISP_SetSmartExposureAttr(ViPipe, &stSmartExpAttr);
```

**【相关主题】**

无。

**4.4.2.15 CVI\_ISP\_GetSmartExposureAttr****【描述】**

获取智能模式下的 AE 曝光属性。仅在智能信息时生效。

**【语法】**

```
CVI_S32 CVI_ISP_GetSmartExposureAttr(VI_PIPE ViPipe, ISP_SMART_EXPOSURE_ATTR_S
→ *pstSmartExpAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstSmartExpAttr	智能模式下的 AE 曝光属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件: cvi\_ae.h
- 库文件: libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetSmartExposureAttr

#### 4.4.2.16 CVI\_ISP\_SetAERouteSFAttr

**【描述】**

设置 AE WDR 短帧曝光分配策略。

**【语法】**

```
CVI_S32 CVI_ISP_SetAERouteSFAttr(VI_PIPE ViPipe, const ISP_AE_ROUTE_S_  
↪ *pstAERouteSFAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteSFAttr	AE WDR 短帧曝光分配策略结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见错误码。

**【需求】**

- 头文件: cvi\_ae.h
- 库文件: libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_GetAERouteSFAttr

- ISP\_AE\_ROUTE\_S

#### 4.4.2.17 CVI\_ISP\_GetAERouteSFAttr

##### 【描述】

获取 AE WDR 短帧曝光分配策略。

##### 【语法】

```
CVI_S32 CVI_ISP_GetAERouteSFAttr(VI_PIPE ViPipe, ISP_AE_ROUTE_S_
→*pstAERouteSFAttr);
```

##### 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteSFAttr	AE WDR 短帧曝光分配策略结构体指针	输出

##### 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

##### 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

##### 【注意】

无。

##### 【举例】

无。

##### 【相关主题】

- CVI\_ISP\_SetAERouteSFAttr

#### 4.4.2.18 CVI\_ISP\_SetAERouteSFAttrEx

##### 【描述】

设置 AE WDR 短帧曝光分配扩展属性，支持分别设置 AE 分配策略中的 sensor 模拟增益，sensor 数字增益和 ISP 数字增益。

##### 【语法】

```
CVI_S32 CVI_ISP_SetAERouteSFAttrEx(VI_PIPE ViPipe, const ISP_AE_ROUTE_EX_S_
→*pstAERouteSFAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteSFAttrEx	AE WDR 短帧曝光分配策略扩展属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetAERouteSFAttrEx](#)
- [ISP\\_AE\\_ROUTE\\_EX\\_S](#)

**4.4.2.19 CVI\_ISP\_GetAERouteSFAttrEx****【描述】**

获取 AE WDR 短帧曝光分配策略扩展属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetAERouteSFAttrEx(VI_PIPE ViPipe, ISP_AE_ROUTE_EX_S_
↪ *pstAERouteSFAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAERouteSFAttrEx	AE WDR 短帧曝光分配策略扩展属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetAERouteSFAttrEx

## 4.4.3 光圈控制模块

### 4.4.3.1 CVI\_ISP\_SetIrisAttr

**【描述】**

设定光圈的控制属性，该函数可实现手动光圈属性和光圈类型等参数的设置。

**【语法】**

```
CVI_S32 CVI_ISP_SetIrisAttr(VI_PIPE ViPipe, const ISP_IRIS_ATTR_S *pstIrisAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIrisAttr	光圈控制属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

- 根据实际对接镜头光圈类型，设置正确的光圈类型属性，由此再去设置相关的 DC-Iris/P-Iris 控制属性。若对接的是手动光圈镜头，可将光圈类型设置为 `ISP_IRIS_DC_TYPE`，建议此时关闭光圈控制使能。
- 手动光圈属性主要用于调试，可通过该 MPI 进行设置。对于 P-Iris 镜头，手动 `enIrisFNO` 值会受到最大、最小光圈目标值的影响。自动光圈属性的更多参数需要调用 `CVI_ISP_SetDcirisAttr` 和 `CVI_ISP_SetPirisAttr` 进行设置。

**【举例】**

无。

**【相关主题】**

- `ISP_IRIS_ATTR_S`
- `CVI_ISP_SetDcirisAttr`
- `CVI_ISP_SetPirisAttr`

#### 4.4.3.2 CVI\_ISP\_GetIrisAttr

**【描述】**

获取光圈的控制属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetIrisAttr(VI_PIPE ViPipe, ISP_IRIS_ATTR_S *pstIrisAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIrisAttr	光圈控制属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件： `cvi_ae.h`
- 库文件： `libae.a`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**



无。

#### 4.4.3.3 CVI\_ISP\_SetDcirisAttr

##### 【描述】

设定 DC-Iris 算法的控制属性，该函数可实现 DC-Iris 自动光圈的参数设置。

##### 【语法】

```
CVI_S32 CVI_ISP_SetDcirisAttr(VI_PIPE ViPipe, const ISP_DCIRIS_ATTR_S *pstDcirisAttr);;
```

##### 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIrisAttr	DC-Iris 自动光圈控制属性结构体指针	输入/输出

##### 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

##### 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

##### 【注意】

DC-Iris 光圈控制采用 PID 算法，算法根据画面亮度，调节 PWM 占空比对光圈大小进行控制。当曝光时间和增益达到最小目标值之后，会进入光圈控制区域。当光圈控制能满足目标亮度的要求时，AE 直接返回，保持曝光时间和增益不变。当画面亮度稳定且 PWM 占空比维持在打开值一段时间后，光圈控制算法会认为光圈已经打开至最大，退出光圈控制区，将控制权交还给 AE

##### 【举例】

无。

##### 【相关主题】

- ISP\_IRIS\_ATTR\_S
- ISP\_DCIRIS\_ATTR\_S

## 4.4.3.4 CVI\_ISP\_GetDcirisAttr

## 【描述】

获取 DC-Iris 自动光圈的控制属性。

## 【语法】

```
CVI_S32 CVI_ISP_GetDcirisAttr(VI_PIPE ViPipe, ISP_DCIRIS_ATTR_S *pstDcirisAttr);;
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIrisAttr	DC-Iris 自动光圈控制属性结构体指针	输出

## 【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

## 【需求】

- 头文件：cvi\_ae.h
- 库文件：libae.a

## 【注意】

无。

## 【举例】

无。

## 【相关主题】

无。

## 4.4.3.5 CVI\_ISP\_SetPirisAttr

## 【描述】

设定 P-Iris 自动光圈的控制属性

## 【语法】

```
CVI_S32 CVI_ISP_SetPirisAttr(VI_PIPE ViPipe, const ISP_PIRIS_ATTR_S *pstPirisAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstPirisAttr	P-Iris 自动光圈控制属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

目前尚未支持 P-iris

**【举例】**

无。

**【相关主题】**

- [ISP\\_PIRIS\\_ATTR\\_S](#)

#### 4.4.3.6 CVI\_ISP\_GetPirisAttr

**【描述】**

获取 P-Iris 自动光圈的控制属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetPirisAttr(VI_PIPE ViPipe, ISP_PIRIS_ATTR_S *pstPirisAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstIrisAttr	P-Iris 自动光圈控制属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_ae.h
- 库文件：libae.a

**【注意】**

目前尚未支持 P-iris

**【举例】**

无。

**【相关主题】**

- [ISP\\_PIRIS\\_ATTR\\_S](#)

## 4.5 数据类型

### 4.5.1 Register

- [AE\\_SENSOR\\_REGISTER\\_S](#)：定义 sensor 注册结构体。
- [AE\\_SENSOR\\_EXP\\_FUNC\\_S](#)：定义 sensor 回调函数结构体。
- [AE\\_SENSOR\\_DEFAULT\\_S](#)：定义 AE 算法库初始化的参数结构体。
- [AE\\_ACCURACY\\_E](#)：定义曝光时间、增益精度类型的枚举。
- [AE\\_ACCURACY\\_S](#)：定义曝光时间、增益精度的结构体。

#### 4.5.1.1 AE\_SENSOR\_REGISTER\_S

**【说明】**

定义 sensor 注册结构体。

**【定义】**

```
typedef struct _AE_SENSOR_REGISTER_S
{
    AE_SENSOR_EXP_FUNC_S stAeExp;
} AE_SENSOR_REGISTER_S;
```

**【成员】**

成员名称	描述
stAeExp	Sensor 注册的回调函数结构体

**【注意事项】**

无。

**【相关数据类型及接口】**

- [AE\\_SENSOR\\_EXP\\_FUNC\\_S](#)

## 4.5.1.2 AE\_SENSOR\_EXP\_FUNC\_S

## 【说明】

定义 sensor 回调函数结构体。

## 【定义】

```
typedef struct _AE_SENSOR_EXP_FUNC_S
{
    CVI_S32(*pfn_cmos_get_ae_default)(VI_PIPE ViPipe, AE_SENSOR_DEFAULT_S_
    ↪ *pstAeSnsDft);
    CVI_VOID(*pfn_cmos_fps_set)(VI_PIPE ViPipe, CVI_FLOAT f32Fps,
    AE_SENSOR_DEFAULT_S *pstAeSnsDft);
    CVI_VOID(*pfn_cmos_slow_framerate_set)(VI_PIPE ViPipe, CVI_U32 u32FullLines, AE_
    ↪ SENSOR_DEFAULT_S *pstAeSnsDft);
    CVI_VOID(*pfn_cmos_inttime_update)(VI_PIPE ViPipe, CVI_U32 *u32IntTime);
    CVI_VOID(*pfn_cmos_gains_update)(VI_PIPE ViPipe, CVI_U32 *u32Again, CVI_U32_
    ↪ *u32Dgain);
    CVI_VOID(*pfn_cmos_again_calc_table)(VI_PIPE ViPipe, CVI_U32 *pu32AgainLin, CVI_U32_
    ↪ *pu32AgainDb);
    CVI_VOID(*pfn_cmos_dgain_calc_table)(VI_PIPE ViPipe, CVI_U32 *pu32DgainLin, CVI_U32_
    ↪ *pu32DgainDb);
    CVI_VOID(*pfn_cmos_get_inttime_max)(VI_PIPE ViPipe, CVI_U16 u16ManRatioEnable, CVI_
    ↪ U32 *au32Ratio, CVI_U32 *au32IntTimeMax, CVI_U32 *au32IntTimeMin, CVI_U32_
    ↪ *pu32LFMaxIntTime);
    CVI_VOID(*pfn_cmos_ae_fswdr_attr_set)(VI_PIPE ViPipe, AE_FSWDR_ATTR_S_
    ↪ *pstAeFSWDRAttr);
} AE_SENSOR_EXP_FUNC_S;
```

## 【成员】

成员名称	描述
pfn_cmos_get_ae_default	获取 AE 算法库初始值的回调函数指针。
pfn_cmos_fps_set	设置 sensor 的帧率。
pfn_cmos_slow_framerate_set	设置 sensor 的降帧。
pfn_cmos_inttime_update	设置 sensor 的曝光时间。
pfn_cmos_gains_update	设置 sensor 的模拟增益和数字增益。
pfn_cmos_again_calc_table	计算 TABLE 类型 sensor 模拟增益。
pfn_cmos_dgain_calc_table	计算 TABLE 类型 sensor 数字增益。
pfn_cmos_get_inttime_max	WDR 模式下，计算短帧最大曝光时间的回调函数指针，与 sensor 强相关。
pfn_cmos_ae_fswdr_attr_set	2to1LineWDR 模式下，设置长帧模式

## 【注意事项】

- 如果回调函数指针不需要赋值，需要置为 NULL。
- pfn\_cmos\_inttime\_update 和 pfn\_cmos\_gains\_update 中设置的曝光时间和增益如何转换成 sensor 的配置值与 sensor 强相关，请参阅 sensor 手册。

## 【相关数据类型及接口】

- AE\_SENSOR\_DEFAULT\_S
- ISP\_SENSOR\_EXP\_FUNC\_S

## 4.5.1.3 AE\_SENSOR\_DEFAULT\_S

## 【说明】

定义 AE 算法库的初始化参数结构体。

## 【定义】

```
typedef struct _AE_SENSOR_DEFAULT_S
{
    CVI_U8 au8HistThresh[HIST_THRESH_NUM];
    CVI_U8 u8AeCompensation;
    CVI_U32 u32LinesPer500ms;
    CVI_U32 u32FlickerFreq;
    CVI_U32 u32HmaxTimes;
    CVI_U32 u32InitExposure;
    CVI_U32 u32InitAESpeed;
    CVI_U32 u32InitAETolerance;
    CVI_U32 u32FullLinesStd;
    CVI_U32 u32FullLinesMax;
    CVI_U32 u32FullLines;
    CVI_U32 u32MaxIntTime;
    CVI_U32 u32MinIntTime;
    CVI_U32 u32MaxIntTimeTarget;
    CVI_U32 u32MinIntTimeTarget;
    AE_ACCURACY_S stIntTimeAccu;
    CVI_U32 u32MaxAgain;
    CVI_U32 u32MinAgain;
    CVI_U32 u32MaxAgainTarget;
    CVI_U32 u32MinAgainTarget;
    AE_ACCURACY_S stAgainAccu;
    CVI_U32 u32MaxDgain;
    CVI_U32 u32MinDgain;
    CVI_U32 u32MaxDgainTarget;
    CVI_U32 u32MinDgainTarget;
    AE_ACCURACY_S stDgainAccu;
    CVI_U32 u32MaxISPDgainTarget;
    CVI_U32 u32MinISPDgainTarget;
    CVI_U32 u32ISPDgainShift;
    CVI_U32 u32MaxIntTimeStep;
    CVI_U32 u32LFMaxShortTime;
    CVI_U32 u32LFMinExposure;
    ISP_AE_STRATEGY_E enAeExpMode;
    CVI_U16 u16ISOCalCoef;
    CVI_U8 u8AERunInterval;
    CVI_FLOAT f32Fps;
    CVI_FLOAT f32MinFps;
    CVI_U32 denom;
    CVI_U32 u32AEResponseFrame;
    CVI_U32 u32SnsStableFrame;
    AE_BLC_TYPE_E enBlcType;
    ISP_SNS_GAIN_MODE_E enWDRGainMode;
} AE_SENSOR_DEFAULT_S;
```

## 【成员】

成员名称	描述
u8AeCompensation	AE 亮度目标值，取值范围为 [0,255]，建议用 0x38~0x40。
u32LinesPer500ms	每 500ms 的总行数。
u32FlickerFreq	抗闪频率，数值为当前电源频率的 256 倍。
u32HmaxTimes	Sensor 读出一行的时间，单位：ns。
u32InitExposure	默认初始曝光量，等于曝光时间 (行数)* 系统增益 (6bit 小数精度)。AE 算法采用该值作为初始 5 帧的曝光量
u32InitAESpeed	默认初始 AE 调节速度，AE 算法采用该值作为初始 100 帧的调节速度
u32InitAETolerance	默认初始 AE 曝光容忍偏差，AE 算法采用该值作为初始 100 帧的曝光容忍偏差值，可用于设置得到首次 AE 收敛稳定标志的亮度范围
u32FullLinesStd	基准帧率时 1 帧的有效行数。
u32FullLinesMax	sensor 降帧后 1 帧所能达到的最大行数，一般设为 sensor 所支持的最大行数。
u32FullLines	sensor 每帧实际生效的总行数。使用 CVI AE 算法时，回调 cmos_fps_set 和 cmos_slow_framerate_set 时必须给该值赋值，用于返回每帧实际生效的总行数。
u32MaxIntTime	最大曝光时间，以行为单位。
u32MinIntTime	最小曝光时间，以行为单位。
u32MaxIntTimeTarget	最大曝光时间目标值，以行为单位。
u32MinIntTimeTarget	最小曝光时间目标值，以行为单位。
stIntTimeAccu	曝光时间精度。
u32MaxAgain	最大模拟增益，以倍为单位。
u32MinAgain	最小模拟增益，以倍为单位。
u32MaxAgainTarget	最大模拟增益目标值，以倍为单位。
u32MinAgainTarget	最小模拟增益目标值，以倍为单位。
stAgainAccu	模拟增益精度。
u32MaxDgain	最大数字增益，以倍为单位。
u32MinDgain	最小数字增益，以倍为单位。
u32MaxDgainTarget	最大数字增益目标值，以倍为单位。
u32MinDgainTarget	最小数字增益目标值，以倍为单位。
stDgainAccu	数字增益精度。
u32MaxISPDgainTarget	最大 ISP 数字增益目标值，以倍为单位。
u32MinISPDgainTarget	最小 ISP 数字增益目标值，以倍为单位。
u32ISPDgainShift	ISP 数字增益精度。
u32MaxIntTimeStep	短帧曝光时间减小的最大调整步长，以行数为单位。仅在两帧合成模式下有效。
u32LFMaxShortTime	自动长帧模式下短帧最大曝光时间。
u32LFMinExposure	自动长帧模式下强制输出长帧的最小曝光量
enAeExpMode	默认曝光策略，高光优先或低光优先。建议 FSWDR 模式设置为低光优先，线性模式及 BuiltInWDR 设置为高光优先。若不设置该值，默认为高光优先。
u16ISOCalCoef	ISO 标定系数，用于保证拍照所需 DCF 信息中显示的 ISO 是标准的，8bit 精度。 取值范围：[0x0, 0xFFFF]， 默认值：为 0x100。

下页继续

表 4.1 – 续上页

成员名称	描述
u8AERunInterval	默认 AE 算法运行间隔，以帧为单位。若不设置，则默认 AE 每帧执行一次。 取值范围：(0x0, 0xFF]
f32Fps	基准帧率
u32AEResponseFrame	Sensor 曝光时间/增益同步生效的所需的帧数

**【注意事项】**

- 线性/WDR 模式切换时，会回调 pfn\_cmos\_get\_ae\_default 函数更新 AE 相关默认参数。若 WDR 模式要使用 AE 扩展分配路线而线性模式不需要，建议在 cmos\_get\_ae\_default 函数里面先对 AE 路线清零：bAERouteExValid = CVI\_FALSE, stAERouteAttr.u32TotalNum = 0, stAERouteAttrEx.u32TotalNum = 0, 然后视需要在 WDR 分支赋值。
- u32LFMaxShortTime 为自动长帧模式下短帧曝光时间的最大值，如果此参数设置过小会导致自动长帧模式下亮区噪声表现变差。

**【相关数据类型及接口】**

- ISP\_SENSOR\_EXP\_FUNC\_S

**4.5.1.4 AE\_ACCURACY\_E****【说明】**

定义曝光时间、增益的精度类型的枚举。

**【定义】**

```
typedef enum _AE_ACCURACY_E
{
    AE_ACCURACY_DB = 0,
    AE_ACCURACY_LINEAR,
    AE_ACCURACY_TABLE,
    AE_ACCURACY_BUTT,
} AE_ACCURACY_E;
```

**【成员】**

成员名称	描述
AE_ACCURACY_DB	精度类型。
AE_ACCURACY_LINEAR	线性精度类型。
AE_ACCURACY_TABLE	表格类型。

**【注意事项】**

无。

**【相关数据类型及接口】**

- AE\_ACCURACY\_S



## 4.5.1.5 AE\_ACCURACY\_S

## 【说明】

定义曝光时间、增益的精度结构体

## 【定义】

```
typedef struct _AE_ACCURACY_S
{
    AE_ACCURACY_E enAccuType;
    float f32Accuracy;
    float f32Offset;
} AE_ACCURACY_S;
```

## 【成员】

成员名称	描述
enAccuType	精度类型，包括线性类型、DB 类型和 TABLE 类型。。
f32Accuracy	精度值。
f32Offset	曝光时间的偏移量，支持正负偏移量设置，以行为单位，该值配置与 sensor 强相关。

## 【注意事项】

无。

## 【相关数据类型及接口】

无。

## 4.5.2 AE

- ISP\_AE\_MODE\_E：定义自动曝光的模式。
- ISP\_AE\_STRATEGY\_E：定义 AE 曝光策略模式。
- ISP\_AE\_DELAY\_S：定义 AE 延时属性。
- ISP\_AE\_RANGE\_S：定义曝光时间或增益的最大值和最小值。
- ISP\_ANTIFLICKER\_MODE\_E：定义抗闪模式。
- ISP\_ANTIFLICKER\_S：定义抗闪属性。
- ISP\_SUBFLICKER\_S：定义 ISP 图像亚抗闪属性。
- ISP\_FSWDR\_MODE\_E：定义 ISP FSWDR 运行模式。
- ISP\_AE\_ATTR\_S：定义自动曝光属性。
- ISP\_ME\_ATTR\_S：定义手动曝光属性。
- ISP\_EXPOSURE\_ATTR\_S：定义 ISP 曝光属性。
- ISP\_WDR\_EXPOSURE\_ATTR\_S：定义 WDR 模式下的曝光属性。
- ISP\_AE\_ROUTE\_NODE\_S：定义 AE 分配路线节点属性。

- `ISP_AE_ROUTE_S`：定义 AE 曝光分配策略属性。
- `ISP_AE_ROUTE_EX_NODE_S`：定义 AE 扩展分配路线节点属性。
- `ISP_AE_ROUTE_EX_S`：定义 AE 曝光分配策略扩展属性。
- `ISP_EXP_INFO_S`：定义 ISP 曝光内部状态信息。

#### 4.5.2.1 ISP\_AE\_MODE\_E

##### 【说明】

定义自动曝光的模式。

##### 【定义】

```
typedef enum _ISP_AE_MODE_E
{
    AE_MODE_SLOW_SHUTTER = 0,
    AE_MODE_FIX_FRAME_RATE = 1,
    AE_MODE_BUTT
} ISP_AE_MODE_E;
```

##### 【成员】

成员名称	描述
<code>AE_MODE_SLOW_SHUTTER</code>	自动降帧模式，即 <code>SLOW_SHUTTER</code> 模式。
<code>AE_MODE_FIX_FRAME_RATE</code>	固定帧率模式。

##### 【注意事项】

- 自动降帧模式是指自动曝光调节时会优先增大曝光时间来尽量减小增益。当 sensor 增益达到用户设置的最大值时，自动曝光调节会逐渐降帧率并延长曝光时间，持续到曝光时间等于自动曝光的最大时间为止。低照度环境下噪声较小但帧率会降低。
- 固定帧率模式是指自动曝光调节时保持帧率不变，低照度环境下噪声会较大。

##### 【相关数据类型及接口】

无。

#### 4.5.2.2 ISP\_AE\_STRATEGY\_E

##### 【说明】

定义 AE 曝光策略模式。

##### 【定义】

```
typedef enum _ISP_AE_STRATEGY_E
{
    AE_EXP_HIGHLIGHT_PRIOR = 0,
    AE_EXP_LOWLIGHT_PRIOR = 1,
    AE_STRATEGY_MODE_BUTT
} ISP_AE_STRATEGY_E;
```

## 【成员】

成员名称	描述
AE_EXP_HIGHLIGHT_PRIORITY	高光优先曝光模式。
AE_EXP_LOWLIGHT_PRIORITY	低光优先曝光模式。

## 【注意事项】

- 线性模式下，AE 算法默认曝光策略为高光优先，尽量避免画面过曝，但在逆光场景时，暗处亮度较低。此时若要关注暗处区域，可采用低光优先模式，但亮处容易过曝。
- WDR 模式下，AE 算法默认使用低光优先模式，控制长帧曝光，此时手动曝光时间、最大最小曝光时间、AE\_Route、AE\_Route\_Ex 中曝光时间参数影响长帧曝光时间。如果切换为高光优先模式，则控制短帧曝光，此时手动曝光时间、最大最小曝光时间、AE\_Route、AE\_Route\_Ex 中曝光时间参数影响短帧曝光时间。

## 【相关数据类型及接口】

无。

## 4.5.2.3 ISP\_AE\_DELAY\_S

## 【说明】

定义 AE 延时属性。

## 【定义】

```
typedef struct _ISP_AE_DELAY_S
{
    CVI_U16 u16BlackDelayFrame;
    CVI_U16 u16WhiteDelayFrame;
} ISP_AE_DELAY_S;
```

## 【成员】

成员名称	描述
u16BlackDelayFrame	图像亮度小于目标亮度时间超过 u 16BlackDelayFrame 帧时，AE 开始调节。
u16WhiteDelayFrame	图像亮度大于目标亮度时间超过 u 16WhiteDelayFrame 帧时，AE 开始调节。

## 【注意事项】

- u16BlackDelayFrame/u16WhiteDelayFrame 设置越大，在 AE 调节步长相同的情况下，调节至目标亮度需要越长时间。人眼对于过曝比较敏感，建议 u16WhiteDelayFrame 要设置得比 u16BlackDelayFrame 小一些。
- 若帧率较低时，建议 u16BlackDelayFrame/u16WhiteDelayFrame 不要设置过大，否则 AE 收敛时间会较长。

## 【相关数据类型及接口】

无。

#### 4.5.2.4 ISP\_AE\_RANGE\_S

**【说明】**

定义曝光时间或增益的最大值和最小值。

**【定义】**

```
typedef struct _ISP_AE_RANGE_S
{
    CVI_U32 u32Max;
    CVI_U32 u32Min;
} ISP_AE_RANGE_S;
```

**【成员】**

成员名称	描述
u32Max	最大值。
u32Min	最小值。

**【注意事项】**

最小值不能大于最大值。

**【相关数据类型及接口】**

无。

#### 4.5.2.5 ISP\_ANTIFLICKER\_MODE\_E

**【说明】**

定义抗闪模式。

**【定义】**

```
typedef enum _ISP_ANTIFLICKER_MODE_E
{
    ISP_ANTIFLICKER_NORMAL_MODE = 0x0,
    ISP_ANTIFLICKER_AUTO_MODE = 0x1,
    ISP_ANTIFLICKER_MODE_BUTT
} ISP_ANTIFLICKER_MODE_E;
```

**【成员】**

成员名称	描述
ISP_ANTIFLICKER_NORMAL_MODE	普通抗闪模式。
ISP_ANTIFLICKER_AUTO_MODE	自动抗闪模式。

**【注意事项】**

- ISP\_ANTIFLICKER\_NORMAL\_MODE 为普通抗闪模式，曝光时间可以根据亮度进行调节，最小曝光时间固定为 1/120 sec (60Hz) 或 1/100 sec(50Hz)，不受曝光时间最小值的限制。

- 有灯光的环境：曝光时间可与光源频率相匹配，能够防止图像闪烁。
- 高亮度的环境：亮度越高，要求曝光时间就越短。而普通抗闪模式的最小曝光时间并不会根据高亮度进行调节，因此会产生过曝情况。
- ISP\_ANTIFLICKER\_AUTO\_MODE 为自动抗闪模式，曝光时间可以根据亮度进行调节，最小曝光时间可达到 sensor 的最小曝光时间。与普通抗闪模式的差别主要在高亮度的环境体现。
- 高亮度的环境：最小曝光时间可以达到 sensor 的最小曝光时间，能够有效抑制过曝，但此时抗闪失效。

#### 【相关数据类型及接口】

无。

#### 4.5.2.6 ISP\_ANTIFLICKER\_S

##### 【说明】

定义抗闪属性。

##### 【定义】

```
typedef struct _ISP_ANTIFLICKER_S
{
    CVI_BOOL bEnable;
    ISP_AE_ANTIFLICKER_FREQUENCY_E enFrequency;
    ISP_ANTIFLICKER_MODE_E enMode;
} ISP_ANTIFLICKER_S;
```

##### 【成员】

成员名称	描述
bEnable	bEnable 为 CVI_TRUE 时使能图像抗闪， 为 CVI_FALSE 时不使能图像抗闪。
enFrequency	抗闪频率值。 取值范围：[0, 1]， 默认值：为 0。0: 60Hz 1: 50Hz
enMode	抗闪模式，普通抗闪或自动抗闪。

##### 【注意事项】

抗闪开启后曝光时间会受到最大/最小曝光时间的限制，若最小抗闪时间大于最大曝光时间，那么开启抗闪后曝光时间将被限制为最大曝光时间。

#### 【相关数据类型及接口】

- ISP\_ANTIFLICKER\_MODE\_E

#### 4.5.2.7 ISP\_SUBFLICKER\_S

**【说明】**

定义 ISP 图像亚抗闪属性。

**【定义】**

```
typedef struct _ISP_SUBFLICKER_S
{
    CVI_BOOL bEnable;
    CVI_U8  u8LumaDiff;
} ISP_SUBFLICKER_S;
```

**【成员】**

成员名称	描述
bEnable	bEnable 为 CVI_TRUE 时使能图像亚抗闪功能，为 CVI_FALSE 时不使能图像亚抗闪。
u8LumaDiff	抗闪程度设置，范围 [0x0, 0x64]。亚抗闪功能生效时，该值越大越接近于抗闪。

**【注意事项】**

- 强制抗闪模式时，最小曝光时间固定为 1/120 sec (60Hz) 或 1/100 sec(50Hz)，在某些场景（如室内对准室外窗户的背光场景）画面可能会过曝严重，但不抗闪画面工频闪又比较厉害。在这种情况下引入亚抗闪模式是为了取得过曝与闪烁之间的平衡。在强制抗闪模式下，当亚抗闪功能生效时，若画面亮度小于 (AeCompensation + u8LumaDiff)，那么曝光时间仍会固定为最小抗闪时间 1/120 sec (60Hz) 或 1/100 sec(50Hz) 以防止图像闪烁。若画面亮度大于 (AeCompensation + u8LumaDiff)，则取消抗闪，调整画面目标亮度为 (AeCompensation + u8LumaDiff)，通过引入一定程度的画面闪烁来避免画面过曝严重。
- 只有在满足打开抗闪、强制抗闪模式且抗闪频率值不等于 0 的前提下，亚抗闪功能才能生效。

**【相关数据类型及接口】**

无。

#### 4.5.2.8 ISP\_FSWDR\_MODE\_E

**【说明】**

定义 ISP FSWDR 运行模式。

**【定义】**

```
typedef enum _ISP_FSWDR_MODE_E
{
    ISP_FSWDR_NORMAL_MODE = 0x0,
    ISP_FSWDR_LONG_FRAME_MODE = 0x1,
    ISP_FSWDR_AUTO_LONG_FRAME_MODE = 0x2,
    ISP_FSWDR_MODE_BUTT
}ISP_FSWDR_MODE_E;
```

## 【成员】

成员名称	描述
ISP_FSWDR_NORMAL_MODE	正常 FSWDR 模式，此时 AE 和合成模块按照自动/手动曝光比进行工作。
ISP_FSWDR_LONG_FRAME_MODE	长帧模式，此时 AE 将短帧曝光时间设置为最小值，长帧曝光时间接近 1 帧所允许的最大值，合成模块只输出长帧数据，可用于优化弱宽动态场景或低照度时的图像质量。
ISP_FSWDR_AUTO_LONG_FRAME_MODE	自动长帧模式，此时 AE 短帧曝光时间不会超过 sensor 限制的长帧模式下短帧最大曝光时间，在曝光量超过 sensor 设置的阈值时，自动切换为长帧模式，合成模块只输出长帧数据。

## 【注意事项】

- 长帧模式仅在行模式 WDR 下有效，自动长帧模式仅在 2to1 行模式 WDR 下有效，并且为了保证图像质量，WDR 模块的运动检测及长短帧融合阈值均为自动配置，不支持手动配置。在线将模式切换到 WDR 模式，默认为正常 FSWDR 模式。
- 自动长帧模式下，手动曝光比生效，如果手动曝光比大于 1，即使曝光量大于设置的阈值也不会进入长帧模式；如果手动曝光比为 1，则会进入长帧模式。

## 【相关数据类型及接口】

无。

## 4.5.2.9 ISP\_AE\_GAIN\_TYPE\_E

## 【说明】

定义增益使用的方式。

## 【定义】

```
typedef enum ISP_AE_GAIN_TYPE_E {
    AE_TYPE_GAIN = 0,
    AE_TYPE_ISO = 1,
    AE_TYPE_BUTT
} ISP_AE_GAIN_TYPE_E;
```

## 【成员】

成员名称	描述
enGainType	0：使用 gain 的方式 1：使用 ISO num 的方式

## 【注意事项】

无。

## 4.5.2.10 ISP\_AE\_ATTR\_S

## 【说明】

定义自动曝光属性。

## 【定义】

```
typedef struct _ISP_AE_ATTR_S
{
    ISP_AE_RANGE_S stExpTimeRange;
    ISP_AE_RANGE_S stAGainRange;
    ISP_AE_RANGE_S stDGainRange;
    ISP_AE_RANGE_S stISPDGainRange;
    ISP_AE_RANGE_S stSysGainRange;
    CVI_U32 u32GainThreshold;
    CVI_U8 u8Speed;
    CVI_U16 u16BlackSpeedBias;
    CVI_U8 u8Tolerance;
    CVI_U8 u8Compensation;
    CVI_U16 u16EVBias;
    ISP_AE_STRATEGY_E enAESTrategyMode;
    CVI_U16 u16HistRatioSlope;
    CVI_U8 u8MaxHistOffset;
    ISP_AE_MODE_E enAEMode;
    ISP_ANTIFLICKER_S stAntiflicker;
    ISP_SUBFLICKER_S stSubflicker;
    ISP_AE_DELAY_S stAEDelayAttr;
    CVI_BOOL bManualExpValue;
    CVI_U32 u32ExpValue;
    ISP_FSWDR_MODE_E enFSWDRMode;
    CVI_BOOL bWDRQuick;
    CVI_U16 u16ISOCalCoef;
    ISP_AE_GAIN_TYPE_E enGainType;
    ISP_AE_RANGE_S stISONumRange;
    CVI_S16 s16IRCutOnLv;
    CVI_S16 s16IRCutOffLv;
    ISP_AE_IR_CUT_FORCE_STATUS enIRCutStatus;
    CVI_U8 au8AdjustTargetMin[LV_TOTAL_NUM];
    CVI_U8 au8AdjustTargetMax[LV_TOTAL_NUM];
    CVI_U16 u16LowBinThr;
    CVI_U16 u16HighBinThr;
    CVI_BOOL bEnableFaceAE;
    CVI_U8 u8FaceTargetLuma;
    CVI_U8 u8FaceWeight;
    CVI_U8 u8GridBvWeight;
    CVI_U32 au32Reserve[RESERVE_SIZE];
    CVI_U8 u8HighLightLumaThr;
    CVI_U8 u8HighLightBufLumaThr;
    CVI_U8 u8LowLightLumaThr;
    CVI_U8 u8LowLightBufLumaThr;
    CVI_BOOL bHistogramAssist;
} ISP_AE_ATTR_S;
```

## 【成员】



成员名称	描述
stExpTimeRange	曝光时间范围, 设置最大值和最小值, 以微秒 (us) 为单位。 取值范围: [0x0, 0xFFFFFFFF], 具体范围与: sensor 相关。
stISONumRange	ISO num 范围, 设置最大值和最小值, 此设定只在 enGainType = AE_TYPE_ISO 时有效。 取值范围: [0x64, 0xFFFFFFFF], 具体范围与: sensor 相关。
stAGainRange	Sensor 模拟增益范围, 设置最大值和最小值, 10bit 小数精度。 取值范围: [0x400, 0xFFFFFFFF], 具体范围与: sensor 相关。
stDGainRange	Sensor 数字增益范围, 设置最大值和最小值, 10bit 小数精度。 取值范围: [0x400, 0xFFFFFFFF], 具体范围与: sensor 相关。
stISPDGainRange	ISP 数字增益范围, 设置最大值和最小值, 10bit 小数精度。 取值范围: [0x400, 0x7FFF]。
stSysGainRange	系统增益范围, 设置最大值和最小值, 10bit 小数精度。 取值范围: [0x400, 0xFFFFFFFF], 具体范围与: sensor 相关。
u32GainThreshold	自动降帧时的系统增益门限值, 10bit 小数精度。 取值范围: [0x400, 0xFFFFFFFF], 默认值: 为 0x400000
u8Speed	自动曝光调整时的速度。 取值范围: [0x0, 0xFF], 默认值: 为 0x40。
u16BlackSpeedBias	画面由暗到亮 AE 调节速度的偏差值, 该值越大, 画面从暗到亮的速度越快。 取值范围: [0x0, 0xFFFF], 默认值: 为 0x90。
u8Tolerance	自动曝光调整时对画面亮度的容忍偏差。 取值范围: [0x0, 0xFF], 默认值: 为 0x2。
u8Compensation	自动曝光调整时的目标亮度。 取值范围: [0x0, 0xFF], 默认值: 为 0x38。
u16EVBias	自动曝光调整时的曝光量偏差值, 10bit 小数精度。 取值范围: [0x0, 0xFFFF], 默认值: 为 0x400。
enAESTrategyMode	自动曝光策略, 高光优先或低光优先, 默认值: 为 AE_EXP_HIGHLIGHT_PRIOR
u16HistRatioSlope	高/低光优先时, 调整目标亮度的步幅。 取值范围: [0x0, 0xFFFF], 默认值: 为 0x8。
u8MaxHistOffset	感兴趣区域对统计平均值影响的最大程度。 取值范围: [0x0, 0xFF], 默认值: 为 0x10。
enAEMode	自动曝光模式, 自动降帧模式或固定帧率模式。 默认值: 为 AE_MODE_FIX_FRAME_RATE
stAntiflicker	抗闪属性设置。默认抗闪不使能。

下页继续

表 4.2 – 续上页

成员名称	描述
stSubflicker	亚抗闪属性设置。默认亚抗闪不使能。
stAEDelayAttr	延时属性设置。默认 u16BlackDelayFrame=0u16WhiteDelayFrame=0。
bManualExpValue	手动曝光量使能，该值为 CVI_TRUE 时，AE 算法采用 u32ExpValue 作为当前曝光量进行曝光时间和增益等的分配，为 CVI_FALSE 时采用自动计算的曝光量进行分配。 默认值：为 CVI_FALSE。
u32ExpValue	手动曝光量值，等于曝光时间 (行数)* 系统增益 (6bit 小数精度)。 默认值：为 0 取值范围：(0x0, 0xFFFFFFFF]。
enFSWDRMode	FSWDR 运行模式。默认为 ISP_FSWDR_NORMAL_MODE。
bWDRQuick	WDR 模式下，AE 算法从稳定状态 (亮度误差小于等于容忍偏差值 u8Tolerance) 重新调整时，默认前 50 帧调整会进行时域滤波，以调整更加平滑。该值为 CVI_TRUE 时，取消 50 帧时域滤波，使 AE 收敛速度更快。默认值为 CVI_FALSE。
u16ISOCalCoef	ISO 标定系数，用于保证拍照所需 DCF 信息中显示的 ISO 是标准的，8bit 精度。 取值范围：[0x0, 0xFFFF]， 默认值：为 0x100。
enGainType	选择以 ISO Num 或者是 Gain 的方式来控制增益
au8AdjustTargetMin	设定环境亮度各 LV AE 收敛的目标亮度的最小值 取值范围：(0x0, 0x100) LV -5 ~ 15 的 默认值：为 { 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45, 50, 50, 50, 50, 50, 60, 60, 60 }
au8AdjustTargetMax	设定环境亮度各 LV AE 收敛的目标亮度的最大值 取值范围：(0x0, 0x100) LV -5 ~ 15 的 默认值：为 { 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 60, 60, 60, 60, 60, 60, 70, 70, 70 }
u16LowBinThr	AE window 统计值低于 lowBinThr(8 bits) 时且累积 window 数低于总 window 数 25%，AE 不使用此 window 的统计值测光， 默认值：为 10
u16HighBinThr	AE window 统计值高于 HighBinThr(8 bits) 时且累积 window 数低于总 window 数 10%，AE 不使用此 window 的统计值测光， 默认值：为 256
s16IRCutOnLv	IR Cut 可切换时，使用 IR Cut 的环境 Lv 值，精度为 100 阈值， 默认值：为 0
s16IRCutOffLv	IR Cut 可切换时，不使用 IR Cut 的环境 Lv 值，精度为 100 默认值：为 700
enIRCutStatus	IR Cut 可切换时，IR Cut 切换的模式设定， 默认值：为 AE_IR_CUT_FORCE_AUTO
bEnableFaceAE	人脸识别连动 AE 测光 (Face AE) 使能
u8FaceTargetLuma	人脸测光的目标亮度 (8 bits)
u8FaceWeight	人脸测光与整体环境测光的权重比例

下页继续

表 4.2 – 续上页

成员名称	描述
u8GridBvWeight	AE 测光分成以 block luma / bvStep 平均这 2 种, 默认以 luma 为主, 此参数可设定 luma 为主的测光混合 bvStep 的测光权重, 高光优先若觉得画像太暗, 可适当调高此参数权重
au32Reserve[RESERVE_SIZE]	保留字段, 无意义
u8HighLightLumaThr	高亮区的亮度阈值
u8HighLightBufLumaThr	高亮缓冲区的亮度阈值
u8LowLightLumaThr	低亮区的亮度阈值
u8LowLightBufLumaThr	低亮缓冲区的亮度阈值
bHistogramAssist	使用直方图统计值来辅助高光策略, 应对高光区域小于一个 block size 的情况

**【注意事项】**

- 自动模式下, 更改手动曝光属性的值不会生效。
- 自动曝光的最大最小时间及增益。

可根据不同的场景对曝光时间及增益进行限定, 如有高速运动物体场景可限定最大曝光时间值为较小值, 这样可减轻运动物体拖影现象。

- 自动曝光的系统增益

若 (sensor 模拟增益最小值 \* sensor 数字增益最小值 \* ISP 数字增益最小值) 小于系统增益最小值, 则 AE 算法内部计算时最小增益会被限制到系统增益的最小值。若 (sensor 模拟增益最大值 \* sensor 数字增益最大值 \* ISP 数字增益最大值) 大于系统增益最大值, 则 AE 算法内部计算时最大增益会被限制到系统增益的最大值。推荐通过设置系统增益的最大、最小值进行增益限制, 分别限制 sensor 模拟增益、sensor 数字增益和 ISP 数字增益时, 若把较高精度的 ISP 数字增益限制到 1 倍, 容易导致闪烁。实际上, AE 算法内部利用系统增益来计算最大/最小曝光量, 而并不是直接限制某项增益的值。比如说系统增益最大值限制为 1 倍, 但 sensor 模拟增益的最小值限制为 2 倍, 则实际生效的结果以 sensor 模拟增益的限制为准, sensor 数字增益及 ISP 数字增益的限制也有同理。

- 自动降帧时的系统增益门限值

在 SLOW\_SHUTTER 模式下, 当系统增益达到所设置的门限值时, 系统将自动进入 SLOW\_SHUTTER 模式。

- u8Speed 用于设定自动曝光时的收敛速度, 该值越大曝光的收敛速度越快, 但也会导致收敛过程中出现反复震荡。
- u16BlackSpeedBias 用于设定画面由暗到亮 AE 调节速度的偏差值。默认 u8Speed 下, 画面从亮到暗的速度会快于画面从暗到亮的速度, 若想双方向的 AE 调节速度差不多, 可以调高 u16BlackSpeedBias。
- 亮度补偿属性 u8Compensation 用于调节曝光的目标亮度。曝光亮度补偿值越大则图像亮度越高。
- 曝光偏差属性 u16EVBias 用于在特殊场景下对画面的目标亮度进行微调, 也可认为是更高精度的亮度补偿值, 通过调节该值来改变画面目标亮度, 真实生效的 AE 目标亮度为  $u8Compensation * u16EVBias / 1024$ 。u8Compensation 不变时, 该值越大则图像亮度越高。
- 曝光容忍偏差属性 u8Tolerance 用于调节曝光对环境的灵敏度, 曝光容忍偏差值越大则曝光越不敏感, 且可能导致同一目标亮度值多次调节得到的亮度差异越大, 所以该属性推荐不能设定过大。

- 曝光策略属性 `enAESTrategyMode` 用于选择对高光优先或低光优先的曝光策略。高光优先意味着对高光敏感, 尽量避免画面过曝。低光优先意味着对低光敏感, 尽量看清楚暗处区域, 不管画面是否过曝。默认的曝光策略是高光优先, 用户可根据场景需要进行调整, 高光优先时, 用户可设定 `u8HighLightLumaThr` 来决定高光区域的亮度阈值, 当画面中有超过此亮度阈值的 windows (该 windows 的权重不为 0) 时, 则会降低目标亮度来抑制此高光 windows 的亮度, 当所有的 windows 的亮度都小 `u8HighLightBufLumaThr` 时, 则恢复到原本设定的目标亮度, 低光优先时, 当画面中有低于此 `u8LowLightLumaThr` 亮度阈值的 windows (该 windows 的权重不为 0) 时, 则会提高目标亮度来拉亮此低光 windows 的亮度, 当所有的 windows 的亮度都大于 `u8LowLightBufLumaThr` 时, 则恢复到原本设定的目标亮度。
- `u16HistRatioSlope` 用于设定感兴趣区域的权重。若是高光优先曝光模式, 则该值设置的是高光区域的权重, 该值越大, 则意味着对高光区域越敏感。反之, 若是低光优先模式, 则该值设置的是低光区域的权重, 该值越大, 则意味着对暗处区域越敏感, 建议 `u16HistRatioSlope` 的值设置不要超过 `0x100`。
- 自动曝光时, 可设定感兴趣区域对统计平均值影响的最大程度 `u8MaxHistOffset`。该值相当于对提高 `u16HistRatioSlope` 时增加的权重做限制, 若该值为 0, 无论 `u16HistRatioSlope` 多大, 也不会对高光或低光区域做特殊处理, 此时的统计平均值就是原始值。通过合理设置该值, 可以保证任何场景 AE 稳定后画面平均亮度都在一定范围内, 在高光优先曝光模式下, 如果该值设置较大, 在对亮度稍高一些的场景, 如晴天室外场景, 有天空有树木, 则可能导致整体画面亮度偏低, 因为此时优先保证了亮区天空的效果, 通过限制该值从而限制对亮区的倾斜程度可以解决该问题。
- 曝光策略切换时, 最好同时更新 `u16HistRatioSlope` 和 `u8MaxHistOffset` 的值, 否则这 2 个值会采用上一种策略下的配置, 效果可能与预期不符。
- 在做强光抑制方案时, 建议在高光优先曝光模式下, 通过降低 AE 目标亮度, 同时合理设置 `u16HistRatioSlope` 和 `u8MaxHistOffset` 来抑制强光, 暗区则可以通过使能 DRC 来看清楚。低光优先则可以用于实现非指定区域的背光补偿。
- AE 曝光控制类型为自动时, 可设定曝光模式 `enAEMode`。该值可设定为慢快门模式 (SLOW\_SHUTTER) 或固定帧率模式。慢快门模式通常用于低照度场景下进行自动降帧, 以减少画面噪声。
- 抗闪属性结构体 `stAntiflicker` 可用于设定抗闪使能, 抗闪频率和抗闪模式等属性。FSWDR 模式下, 只对长帧抗闪有效。由于随着曝光比的变化, 短帧曝光时间的最大值会受到限制, 建议 FSWDR 模式使用自动抗闪模式。
- 亚抗闪属性结构体 `stSubflicker` 可用于设定亚抗闪使能及抗闪程度属性。若有自动光圈, 建议关闭亚抗闪功能。
- AE 曝光控制类型为自动时, 可设定 AE 延时生效属性结构体 `stAEDelayAttr`。该值的合理设置可提高画面亮度的稳定性, 防止快速运动物体经过导致画面亮度发生变化。低码率设置时应该适当增加该值, 以避免 AE 调节时出现块效应。
- 用户可以通过将 `bManualExpValue` 置为 `CVI_TRUE`, 手动设置曝光量 `u32ExpValue` 来屏蔽 AE 算法的曝光量调整部分, 只用到曝光量分配部分。`u32ExpValue` 的值会受到最大、最小曝光量的限制。最大、最小曝光量分别对应最大、最小曝光时间与增益的乘积。
- FSWDR 运行模式改变时, 需要相应修改 `cmos.c` 中的以下回调函数:

`cmos_get_ae_default`, `cmos_fps_set`, `cmos_get_inttime_max`, 以保证长、短帧曝光时间在合理范围内。正常 WDR 和长帧模式切换时, 会回调 `cmos_get_ae_default`, 更新曝光时间最大/最小值, 增益最大/最小值, 光圈最大/最小值和 AE compensation 等参数, 同时, AE 内部会

将 u16HistRatioSlope 和 u8MaxHistOffset 更新为算法默认值。需要注意的是，此时曝光比相关设置不会更新，仍会保持上一次的值。

- au8AdjustTargetMin/ au8AdjustTargetMax 可针对不同的环境亮度设定不同的目标亮度最小/大值, AE 会根据当前的环境亮度收敛目标亮度到该区间内
- u16LowBinThr/ u16HighBinThr 可设定该参数来排除 AE 测光时, 影像中太暗及太亮的部分, 建议 u16LowBinThr 不要超过 10( 8 bits)
- enIRCutStatus/ s16IRCutOnLv/ s16IRCutOffLv 当产品的 IR Cut 为可切换时, 该参数才会有作用, 切换模式为 Auto 时, 会根据设定的 On/Off 的环境亮度来切换 IR Cur, 建议这 2 个环境亮度差要超过 6 个 LV 以上, 才不会发生频繁的反复切换的情况

#### 【相关数据类型及接口】

- CVI\_ISP\_SetExposureAttr

#### 4.5.2.11 ISP\_ME\_ATTR\_S

##### 【说明】

定义手动曝光属性。

##### 【定义】

```
typedef struct _ISP_ME_ATTR_S
{
    ISP_OP_TYPE_E enExpTimeOpType;
    ISP_OP_TYPE_E enAGainOpType;
    ISP_OP_TYPE_E enDGainOpType;
    ISP_OP_TYPE_E enISPDGainOpType;
    CVI_U32 u32ExpTime;
    CVI_U32 u32AGain;
    CVI_U32 u32DGain;
    CVI_U32 u32ISPDGain;
    ISP_OP_TYPE_E enISONumOpType;
    ISP_AE_GAIN_TYPE_E enGainType;
    CVI_U32 u32ISONum;
} ISP_ME_ATTR_S;
```

##### 【成员】



成员名称	描述
enExpTimeOpType	手动曝光时间使能， 默认值：为 OP_TYPE_AUTO
enAGainOpType	手动 sensor 模拟增益使能， 默认值：为 OP_TYPE_AUTO
enDGainOpType	手动 sensor 数字增益使能， 默认值：为 OP_TYPE_AUTO
enISPDGainOpType	手动 ISP 数字增益使能， 默认值：为 OP_TYPE_AUTO
u32ExpTime	手动曝光时间，以微秒 (us) 为单位， 默认值：为 0x4000。 取值范围：[0x0, 0xFFF FFFF]， 具体范围与：sensor 相关。
u32AGain	手动 sensor 模拟增益，10bit 小数精度， 默认值：为 0x400。 取值范围：[0x400, 0xFFF FFFF]， 具体范围与：sensor 相关。
u32DGain	手动 sensor 数字增益，10bit 小数精度， 默认值：为 0x400。 取值范围：[0x400, 0xFFF FFFF]， 具体范围与：sensor 相关。
u32ISPDGain	手动 ISP 数字增益，10bit 小数精度， 默认值：为 0x400。 取值范围：[0x400, 0x 40000]， 具体范围与：sensor 相关。
enISONumOpType	选择以 ISO Num 或 gain 的方式控制增益， 默认值：为 0
enGainType	增益使用的方式
u32ISONum	手动 ISO Num 增益， 默认值：为 100，只在 enGainType = AE_TYPE_ISO 有作用。

**【注意事项】**

- 手动模式下，手动曝光时间和增益的值会受到自动模式下最大/最小曝光时间和增益的限制。
- 手动曝光使能参数有效时，必须设置相应的手动曝光参数，若不设置，则采用系统默认值：。
- 增益单位为 10bit 小数精度的倍数，即 1024 代表 1 倍。
- 若曝光参数设置超出最大（小）值，将使用 sensor 支持的最大（小）值代替。

**【相关数据类型及接口】**

- [CVI\\_ISP\\_SetExposureAttr](#)

## 4.5.2.12 ISP\_EXPOSURE\_ATTR\_S

## 【说明】

定义 ISP 曝光属性。

## 【定义】

```
typedef struct _ISP_EXPOSURE_ATTR_S
{
    CVI_BOOL bByPass;
    ISP_OP_TYPE_E enOpType;
    CVI_U8 u8AERunInterval;
    CVI_BOOL bHistStatAdjust;
    CVI_BOOL bAERouteExValid;
    ISP_ME_ATTR_S stManual;
    ISP_AE_ATTR_S stAuto;
    CVI_U8 u8DebugMode;
    ISP_AE_METER_MODE_E enMeterMode;
    CVI_BOOL bAEGainSepCfg;
} ISP_EXPOSURE_ATTR_S;
```

## 【成员】

成员名称	描述
bByPass	AE 模块 bypass 功能使能，默认为 CVI_FALSE。
enOpType	自动曝光或手动曝光开关，默认为 OP_TYPE_AUTO。
u8AERunInterval	AE 算法运行的间隔，取值范围为 [1,255]。 取值为 1 时表示每帧都运行 AE 算法；取值为 2 时表每 2 帧运行 1 次 AE 算法，依此类推。 建议该值设置不要大于 2， 否则 AE 调节速度会受到影响。 WDR 模式时，该值建议设置为 1，这样 AE 收敛会更加平滑。 该值默认为 1。
bHistStatAdjust	此参数会压低 AE 对于亮区的测光亮度，使亮区更亮，对于大面积天空场景，若测光偏暗，可设定此参数默认为 0
bAERouteExValid	AE 扩展分配路线是否生效开关 CVI_TRUE 时使用 AE 扩展分配路线，否则使用普通 AE 分配路线。 默认为 CVI_FALSE。
stManual	手动曝光属性结构体
stAuto	自动曝光属性结构体
u8DebugMode	设置 AE 的 debug mode，用于 dumplog 等 debug 功能，正常为 0
enMeterMode	AE 测光方式 0：AE_METER_MULTI 中央加权测光 1：AE_METER_AVERAGE 平均加权测光 2：AE_METER_HIGHLIHGT_PRIORITY 亮区优先测光， 适合 IR sensor 使用， 默认值：为 AE_METER_MULTI
bAEGainSepCfg	长短帧增益是否分开分配

**【注意事项】**

- AE ByPass 为 CVI\_TRUE 时，AE 模块被 bypass，任何 AE 配置都不会对图像亮度产生影响。ISP\_AE\_RESULT\_S ‘\_’ 保持为 AE bypass 前一帧的值。
- WDR/线性模式切换时，u8AERunInterval 会重置为 1，用户可以在切换完成后，根据需要修改该值。
- bAEGainSepCfg 仅在 2to1WDR 模式下配置且 sensor 支持此功能才能生效。
- 手动模式不支持长短帧增益分开分配

**【相关数据类型及接口】**

- CVI\_ISP\_SetExposureAttr
- CVI\_ISP\_GetExposureAttr

**4.5.2.13 ISP\_WDR\_EXPOSURE\_ATTR\_S****【说明】**

定义 WDR 模式下的曝光属性。

**【定义】**

```
#define EXP_RATIO_NUM      (3)
typedef struct _ISP_WDR_EXPOSURE_ATTR_S
{
    ISP_OP_TYPE_E enExpRatioType;
    CVI_U32 au32ExpRatio[EXP_RATIO_NUM];
    CVI_U32 u32ExpRatioMax;
    CVI_U32 u32ExpRatioMin;
    CVI_U16 u16Tolerance;
    CVI_U16 u16Speed;
    CVI_U16 u16RatioBias;
    CVI_U8 u8SECompensation;
    CVI_U16 u16SEHisThr;
    CVI_U16 u16SEHisCntRatio1;
    CVI_U16 u16SEHisCntRatio2;
    CVI_U16 u16SEHis255CntThr1;
    CVI_U16 u16SEHis255CntThr2;
    CVI_U8 au8LEAdjustTargetMin[LV_TOTAL_NUM];
    CVI_U8 au8LEAdjustTargetMax[LV_TOTAL_NUM];
    CVI_U8 au8SEAdjustTargetMin[LV_TOTAL_NUM];
    CVI_U8 au8SEAdjustTargetMax[LV_TOTAL_NUM];
    CVI_U8 u8AdjustTargetDetectFrmNum;
    CVI_U32 u32DiffPixelNum;
    CVI_U16 u16LELowBinThr;
    CVI_U16 u16LEHighBinThr;
    CVI_U16 u16SELowBinThr;
    CVI_U16 u16SEHighBinThr;
    CVI_U8 au8FrameAvgLumaMin[LV_TOTAL_NUM];
    CVI_U8 au8FrameAvgLumaMax[LV_TOTAL_NUM];
} ISP_WDR_EXPOSURE_ATTR_S;
```

**【成员】**



成员名称	描述
enExpRatioType	<p>仅在多帧合成 WDR 模式下有效。</p> <p>OP_TYPE_AUTO: 根据场景自动计算长短帧曝光比;</p> <p>OP_TYPE_MANUAL: 手动配置长短帧曝光比。</p>
au32ExpRatio	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, au32ExpRatio 无效。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, au32ExpRatio 表示二帧合成 WDR 相邻 2 帧曝光比期望值。</p> <p>其中 au32ExpRatio[0] 作为长帧/短帧曝光比, 6bit 小数精度, 0x40 表示曝光比为 1 倍。</p> <p>取值范围: [0x40, 0xFFFF]</p>
u32ExpRatioMax	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMax 表示最长帧与最短帧曝光时间比值的最大值。</p> <p>即 2 帧合成时表示长帧/短帧曝光比的最大值。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMax 无效。6bit 小数精度, 0x40 表示曝光比为 1 倍。</p> <p>取值范围: [0x40, 0x4000]</p>
u32ExpRatioMin	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMin 表示长帧曝光时间与短帧曝光时间比值的最小值。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMin 无效。</p> <p>格式为无符号 6.6bit 定点, 0x40 表示长帧曝光时间与短帧曝光时间的比值为 1 倍。</p> <p>默认值: 为 0x40。</p> <p>取值范围: [0x40, u32ExpRatioMax]</p>
u16Tolerance	<p>曝光调整时对画面亮度的容忍偏差, 仅在两帧合成 WDR 模式下有效。</p> <p>默认值: 为 5 取值范围: [0x0, 0xFF]</p>
u16Speed	<p>自动曝光比调节速度, 仅在两帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 自动曝光比调节速度越快。</p> <p>默认值: 为 0x20。</p> <p>取值范围: [0x0, 0xFF]</p>
u16RatioBias	<p>曝光比偏差值, 仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 自动曝光比越大。</p> <p>默认值: 为 0x400, 表示不对自动曝光比算法的计算结果进行调整。</p> <p>经过该值调整的曝光比会受到曝光比最大/最小值的限制。</p> <p>取值范围: [0x0, 0xFFFF]</p>
u8SECompensation	<p>短帧的目标亮度</p> <p>默认值: 为 56</p>
au8LEAdjustTargetMin	<p>设定长帧环境亮度各 LV AE 收敛的目标亮度的最小值</p> <p>取值范围: (0x0, 0x100), LV 5 ~ 15 的</p>
	<p>默认值: 为 {14, 15, 15, 15, 15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60, 60, 60, 60}</p>
au8LEAdjustTargetMax	<p>设定长帧环境亮度各 LV AE 收敛的目标亮度的最大值</p>

## 【注意事项】

- 针对某些对短帧最大曝光时间有限制的 sensor，曝光比较小时，长帧最大曝光时间较短，图像动态范围及噪声表现较差，导致自动曝光比计算不准确，此时建议限制最小曝光比，保证长帧最大曝光时间至少达到 3ms。
- 建议 u16Speed 不要设置小于 0x8，避免某些场景由于计算精度不足导致曝光比调节太慢甚至不调。u16Speed 过大可能导致曝光比变化太快，造成画面亮度出现振荡。
- 建议手动模式下 u32ExpRatio 不要设置大于 0x400。如果曝光比大于 0x400，在较亮的超宽动态场景，适当调高曝光比会优化长帧图像噪声表现。但是在较暗或低宽动态场景，如果曝光比过大，由于短帧最大曝光时间会被压缩，导致图像噪声表现会变差，而且会出现明显的噪声不连续，运动表现也会变差。

## 【相关数据类型及接口】

- CVI\_ISP\_SetWDRExposureAttr
- CVI\_ISP\_GetWDRExposureAttr

## 4.5.2.14 ISP\_AE\_ROUTE\_NODE\_S

## 【说明】

定义 AE 分配路线节点属性。

## 【定义】

```
typedef struct _ISP_AE_ROUTE_NODE_S
{
    CVI_U32 u32IntTime;
    CVI_U32 u32SysGain;
    ISP_IRIS_F_NO_E enIrisFNO;
    CVI_U32 u32IrisFNOLin;
} ISP_AE_ROUTE_NODE_S;
```

## 【成员】

成员名称	描述
u32IntTime	节点曝光时间，单位为微秒 (us)。 取值范围：(0x0, 0xFFFFFFFF]
u32SysGain	节点增益，包括 sensor 模拟增益，sensor 数字增益和 ISP 数字增益，10bit 精度。 取值范围：[0x400, 0xFFFFFFFF]
enIrisFNO	节点光圈 F 值大小，仅支持 P-Iris，不支持 DC-Iris。 取值范围：[ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]。
u32IrisFNOLin	节点光圈 F 值等效增益大小，仅支持 P-Iris，不支持 DC-Iris。 取值范围：[1, 1024]

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetAERouteAttr

#### 4.5.2.15 ISP\_AE\_ROUTE\_S

##### 【说明】

定义 AE 曝光分配策略属性。

##### 【定义】

```
#define ISP_AE_ROUTE_MAX_NODES (16)
typedef struct _ISP_AE_ROUTE_S
{
    CVI_U32 u32TotalNum;
    ISP_AE_ROUTE_NODE_S astRouteNode[ISP_AE_ROUTE_MAX_NODES];
} ISP_AE_ROUTE_S;
```

##### 【成员】

成员名称	描述
u32TotalNum	曝光分配路线节点数目，目前最大为 16。
astRouteNode[ISP_AE_ROUTE_MAX_NODES]	曝光分配路线节点属性。

##### 【注意事项】

无。

##### 【相关数据类型及接口】

- CVI\_ISP\_SetAERouteAttr

#### 4.5.2.16 ISP\_AE\_ROUTE\_EX\_NODE\_S

##### 【说明】

定义 AE 扩展分配路线节点属性。

##### 【定义】

```
typedef struct _ISP_AE_ROUTE_EX_NODE_S
{
    CVI_U32 u32IntTime;
    CVI_U32 u32Again;
    CVI_U32 u32Dgain;
    CVI_U32 u32IspDgain;
    ISP_IRIS_F_NO_E enIrisFNO;
    CVI_U32 u32IrisFNOLin;
} ISP_AE_ROUTE_EX_NODE_S;
```

##### 【成员】

成员名称	描述
u32IntTime	节点曝光时间，单位为微秒 (us)。 取值范围：(0x0, 0xFFFFFFFF]
u32Again	sensor 模拟增益，10bit 精度。 取值范围：[0x400, 0x3FFFFFF]
u32Dgain	sensor 数字增益，10bit 精度。 取值范围：[0x400, 0x3FFFFFF]
u32IspDgain	ISP 数字增益，10bit 精度。 取值范围：[0x400, 0x40000]
enIrisFNO	节点光圈 F 值大小，仅支持 P-Iris，不支持 DC-Iris。 取值范围：[ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]
u32IrisFNOLin	节点光圈 F 值等效增益大小，仅支持 P-Iris，不支持 DC-Iris。 取值范围：[1, 1024]

**【注意事项】**

无。

**【相关数据类型及接口】**

· CVI\_ISP\_SetAERouteAttrEx

**4.5.2.17 ISP\_AE\_ROUTE\_EX\_S****【说明】**

定义 AE 曝光分配策略扩展属性。

**【定义】**

```
#define ISP_AE_ROUTE_EX_MAX_NODES (16)
typedef struct _ISP_AE_ROUTE_EX_S
{
    CVI_U32 u32TotalNum;
    ISP_AE_ROUTE_EX_NODE_S astRouteExNode[ISP_AE_ROUTE_EX_MAX_NODES];
} ISP_AE_ROUTE_EX_S;
```

**【成员】**

成员名称	描述
u32TotalNum	曝光扩展分配路线节点数目，目前最大为 16。
astRouteExNode[ISP_AE_ROUTE_EX_MAX_NODES]	曝光扩展分配路线节点属性。

**【注意事项】**

无。

**【相关数据类型及接口】**

· CVI\_ISP\_SetAERouteAttrEx

## 4.5.2.18 ISP\_EXP\_INFO\_S

## 【说明】

定义 ISP 曝光内部状态信息。

## 【定义】

```
#define HIST_NUM (256)
typedef struct _ISP_EXP_INFO_S
{
    CVI_U32 u32ExpTime;
    CVI_U32 u32ShortExpTime;
    CVI_U32 u32MedianExpTime;
    CVI_U32 u32LongExpTime;
    CVI_U32 u32AGain;
    CVI_U32 u32DGain;
    CVI_U32 u32ISPDGain;
    CVI_U32 u32Exposure;
    CVI_BOOL bExposureIsMAX;
    CVI_S16 s16HistError;
    CVI_U32 au32AE_Hist256Value[HIST_NUM];
    CVI_U8 u8AveLum;
    CVI_U32 u32LinesPer500ms;
    CVI_U32 u32PirisFNO;
    CVI_U32 u32Fps;
    CVI_U32 u32ISO;
    CVI_U32 u32ISOCalibrate;
    CVI_U32 u32RefExpRatio;
    CVI_U32 u32FirstStableTime;
    ISP_AE_ROUTE_S stAERoute;
    ISP_AE_ROUTE_EX_S stAERouteEx;
    CVI_U8 u8WDRShortAveLuma;
    CVI_U32 u32WDRExpRatio;
    CVI_U8 u8LEFrameAvgLuma;
    CVI_U8 u8SEFrameAvgLuma;
    CVI_FLOAT fLightValue;
    CVI_U32 u32AGainSF;
    CVI_U32 u32DGainSF;
    CVI_U32 u32ISPDGainSF;
    CVI_U32 u32ISOSF;
    ISP_AE_ROUTE_S stAERouteSF;
    ISP_AE_ROUTE_EX_S stAERouteSFEx;
    CVI_BOOL bGainSepStatus;
}ISP_EXP_INFO_S;
```

## 【成员】

成员名称	描述
u32ExpTime	当前曝光时间，单位为微秒 (us)。 取值范围：[0x0, 0xFFFFFFFF]
u32ShortExpTime	FSW DR 模式下，表示当前短帧 (S) 曝光时间，单位为微秒 (us)。线性模式不用关注该值。 取值范围：[0x0, 0xFFFFFFFF]
u32LongExpTime	FSWDR 模式下，表示当前长帧曝光时间，单位为微秒 (us)。 取值范围：[0x0, 0xFFFFFFFF]
u32AGain	当前 sensor 模拟增益，10bit 小数精度。 取值范围：[0x400, 0xFFFFFFFF]
u32DGain	当前 sensor 数字增益，10bit 小数精度。 取值范围：[0x400, 0xFFFFFFFF]
u32ISPDGain	当前 ISP 数字增益，10bit 小数精度。 取值范围：[0x400, 0xFFFFFFFF]
u32Exposure	当前曝光量，等于曝光时间与曝光增益的乘积，其中曝光时间的单位为曝光行数，曝光增益为 6bit 小数精度。 取值范围：[0x40, 0xFFFFFFFF]
bExposureIsMAX	0: ISP 未达到最大曝光水平；1: ISP 达到最大曝光水平。
s16HistError	统计信息，AE 的目标亮度值与实际值的差，该值为正表示当前期望的亮度信息大于实际的亮度信息，该值为负表示期望的亮度信息小于实际的亮度信息。
au32AE_Hist256Value	全局 256 段直方图统计信息取值范围：[0x0, 0xFFFFFFFF]
u8AveLum	平均亮度信息取值范围：[0x0, 0xFF]
u32LinesPer500ms	当前每 500ms 对应的曝光行数，可用于将曝光时间的单位由 us 转换成行数。 取值范围：[0x0, 0xFFFFFFFF]
u32PirisFNO	当前 P-Iris 光圈 F 值对应的等效增益。取值范围：[0x0, 0x400]
u32Fps	实际图像帧率 * 100 取值范围：[0x0, 0xFFFFFFFF]
u32ISO	当前 sensor 模拟增益 * sensor 数字增益 * ISP 数字增益 * 100，其中增益的精度都为 10bit。 取值范围：[0x64, 0xFFFFFFFF]
u32ISOCalibrate	标准 ISO，用于拍照 DCF 信息显示。u32ISOCalibrate = u32ISO * 256 / u16ISOCalCoef。
u32RefExpRatio	参考曝光比，用于估计当前场景的动态范围，会受到 ISP_WDR_EXPOSURE_ATTR_S 中 Tolerance 和 Speed 等值的影响。 取值范围：[0x40, 0x4000]
u32FirstStableTime	首次 AE 收敛稳定的时间，单位为微秒 (us)。 取值范围：[0x0, 0xFFFFFFFF]
stAERoute	实际生效的 AE route，各个节点中的曝光时间以 us 为单位，增益为 10bit 精度，光圈取值范围为 [ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]。光圈类型为 DC-Iris 时，节点光圈值不会对曝光量分配产生影响。
stAERouteEx	实际生效的扩展 AE route，各个节点中的曝光时间以 us 为单位，增益为 10bit 精度，光圈取值范围为 [ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]。光圈类型为 DC-Iris 时，节点光圈值不会对曝光量分配产生影响。
u32WDRExpRatio	WDR 长/短帧的曝光比，6 bits 精度
u8WDRShortAveLuma	短帧的画面亮度
u8LEFrameAvgLuma	长帧的画面平均亮度
u8SEFrameAvgLuma	短帧的画面平均亮度
fLightValue	环境亮度值

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_QueryExposureInfo

**4.5.2.19 ISP\_SMART\_EXPOSURE\_ATTR\_S****【说明】**

定义智能模式下的 AE 曝光属性。

**【定义】**

```
typedef struct _ISP_SMART_EXPOSURE_ATTR_S
{
    CVI_BOOL bEnable;
    CVI_BOOL bIRMode;
    ISP_OP_TYPE_E enSmartExpType;
    CVI_U8 u8LumaTarget;
    CVI_U16 u16ExpCoef;
    CVI_U16 u16ExpCoefMax;
    CVI_U16 u16ExpCoefMin;
    CVI_U8 u8SmartInterval;
    CVI_U8 u8SmartSpeed;
    CVI_U16 u16SmartDelayNum;
    CVI_U8 u8Weight;
    CVI_U8 u8NarrowRatio;
} ISP_SMART_EXPOSURE_ATTR_S;
```

**【成员】**

成员名称	描述
bEnable	智能 AE 是否使能，默认 CVI_FALSE。
bIRMode	是否为红外模式，默认 CVI_FALSE。CVI_TRUE 为红外模式；。CVI_FALSE 为普通模式。
enSmartExpType	OP_TYPE_AUTO：根据目标检测结果自动调整曝光系数； OP_TYPE_MANUAL：手动配置曝光系数。
u8LumaTarget	识别区域 (如人脸) 目标亮度取值范围：[0x0, 0xFF]
u16ExpCoef	基于原始曝光调整的曝光系数， 值越大，图像整体亮度越高； 值越小，图像整体亮度越低。 当 enSmartExpType 为 OP_TYPE_AUTO 时，u16ExpCoef 为 AE 算法根据目标检测结果计算的曝光系数。 当 enExpRatioType 为 OP_TYPE_MANUAL 时，u16ExpCoef 为手动配置的曝光系数。 10bit 小数精度，0x400 表示曝光系数为 1 倍。 取值范围：[0x0, 0xFFFF]
u16ExpCoefMax	智能曝光系数最大值。 当 enSmartExpType 为 OP_TYPE_AUTO 时，u16ExpCoefMax 表示智能曝光系数最大值。 当 enSmartExpType 为 OP_TYPE_MANUAL 时，u16ExpCoefMax 无效。 10bit 小数精度，0x400 表示曝光系数最大值为 1 倍。 取值范围：[0x0, 0xFFFF]
u16ExpCoefMin	智能曝光系数最小值。 当 enSmartExpType 为 OP_TYPE_AUTO 时，u16ExpCoefMin 表示。智能曝光系数最大值。 当 enSmartExpType 为 OP_TYPE_MANUAL 时，u16ExpCoefMin 无效。 10bit 小数精度，0x400 表示曝光系数最小值为 1 倍。 取值范围：[0x0, 0xFFFF]
u8SmartInterval	智能 AE 运行间隔。 取值为 1 时表示每帧运行， 取值为 n 时，表示每 n 帧运行一次。 取值范围：[0x1, 0xFF]
u8SmartSpeed	智能 AE 调整速度。 默认值：为 0x40，值越小，调整速度越慢；值越大，调整速度越快，也越容易出现震荡。 取值范围：[0x0, 0xFF]
u16SmartDelayNum	智能 AE 延时帧数，取值为 4 时表示 4 帧未设置识别区域信息，清空检测区域信息 ISP_SMART_INFO_S。
u8Weight	识别区域运算权重，仅智能模式生效 取值范围：[0x0, 0x64]
u8NarrowRatio	识别区域有效面积缩小比例 取值范围：[0x0, 0x64]

## 【注意事项】

无。

## 【相关数据类型及接口】



- CVI\_ISP\_SetSmartExposureAttr
- CVI\_ISP\_GetSmartExposureAttr

### 4.5.3 IRIS

- ISP\_IRIS\_STATUS\_E：定义 ISP 光圈状态。
- ISP\_IRIS\_TYPE\_E：定义 ISP 光圈类型。
- ISP\_IRIS\_F\_NO\_E：定义 ISP 光圈 F 值。
- ISP\_MI\_ATTR\_S：定义手动光圈属性。
- ISP\_DCIRIS\_ATTR\_S：定义 DC-Iris 算法属性。
- ISP\_PIRIS\_ATTR\_S：定义 P-Iris 属性。
- ISP\_IRIS\_ATTR\_S：定义 ISP 光圈属性。

#### 4.5.3.1 ISP\_IRIS\_STATUS\_E

##### 【说明】

定义 ISP 光圈状态。

##### 【定义】

```
typedef enum _ISP_IRIS_STATUS_E
{
    ISP_IRIS_KEEP = 0,
    ISP_IRIS_OPEN = 1,
    ISP_IRIS_CLOSE = 2,
    ISP_IRIS_BUTT
} ISP_IRIS_STATUS_E;
```

##### 【成员】

成员名称	描述
ISP_IRIS_KEEP	光圈保持当前状态。
ISP_IRIS_OPEN	光圈全开。
ISP_IRIS_CLOSE	光圈全关。

##### 【注意事项】

该值设置为 ISP\_IRIS\_OPEN 或 ISP\_IRIS\_CLOSE 时，光圈处于全开或全关的状态，可用于测试光圈控制电路和驱动是否正确。OPEN 和 CLOSE 的优先级高于光圈控制使能和手动/自动模式，光圈控制算法运行时，为保证其能够正常工作，需将该值设置为 ISP\_IRIS\_KEEP。

##### 【相关数据类型及接口】

无。

## 4.5.3.2 ISP\_IRIS\_TYPE\_E

## 【说明】

定义 ISP 光圈类型。

## 【定义】

```
typedef enum _ISP_IRIS_TYPE_E
{
    ISP_IRIS_DC_TYPE = 0,
    ISP_IRIS_P_TYPE,
    ISP_IRIS_TYPE_BUTT,
} ISP_IRIS_TYPE_E;
```

## 【成员】

成员名称	描述
ISP_IRIS_DC_TYPE	DC-Iris 光圈
ISP_IRIS_P_TYPE	P-Iris 光圈

## 【注意事项】

- 必须设置正确的光圈类型，光圈控制算法才能正常工作。
- 若对接的是手动光圈镜头，可将该值设置为 ISP\_IRIS\_DC\_TYPE，建议此时关闭光圈控制使能。

## 【相关数据类型及接口】

无。

## 4.5.3.3 ISP\_IRIS\_F\_NO\_E

## 【说明】

定义 ISP 光圈 F 值。

## 【定义】

```
typedef enum _ISP_IRIS_F_NO_E
{
    ISP_IRIS_F_NO_32_0 = 0,ISP_IRIS_F_NO_22_0,ISP_IRIS_F_NO_16_0,ISP_IRIS_F_NO_
    ↪11_0,ISP_IRIS_F_NO_8_0,ISP_IRIS_F_NO_5_6,ISP_IRIS_F_NO_4_0,ISP_IRIS_F_NO_
    ↪2_8,ISP_IRIS_F_NO_2_0,ISP_IRIS_F_NO_1_4,ISP_IRIS_F_NO_1_0,ISP_IRIS_F_NO_
    ↪BUTT,
} ISP_IRIS_F_NO_E;
```

## 【成员】

成员名称	描述	等效增益
ISP_IRIS_F_NO_32	光圈 F32.0	1
ISP_IRIS_F_NO_22	光圈 F22.0	2
ISP_IRIS_F_NO_16	光圈 F16.0	4
ISP_IRIS_F_NO_11	光圈 F11.0	8
ISP_IRIS_F_NO_8	0光圈 F8.0	16
ISP_IRIS_F_NO_5	6光圈 F5.6	32
ISP_IRIS_F_NO_4	0光圈 F4.0	64
ISP_IRIS_F_NO_2	8光圈 F2.8	128
ISP_IRIS_F_NO_2	0光圈 F2.0	256
ISP_IRIS_F_NO_1	4光圈 F1.4	512
ISP_IRIS_F_NO_1	0光圈 F1.0	1024

**【注意事项】**

- 针对 P-Iris, AE 算法根据分配路线计算曝光量时, 光圈 F 值要等效成一个增益, 公式如下:  
等效增益  $FNO = 1 \llcorner ISP\_IRIS\_F\_NO\_XX\_XX$ 。由此可知 F32.0 对应增益 1, F22.0 对应增益 2, F16.0 对应增益 4, 以此类推, F1.0 对应增益 1024。

**【相关数据类型及接口】**

无。

**4.5.3.4 ISP\_MI\_ATTR\_S****【说明】**

定义手动光圈属性。

**【定义】**

```
typedef struct _ISP_MI_ATTR_S
{
    CVI_U32 u32HoldValue;
    ISP_IRIS_F_NO_E enIrisFNO;
} ISP_MI_ATTR_S;
```

**【成员】**

成员名称	描述
u32HoldValue	光圈校正值, 用于 DC-Iris 的调试。取值范围为 [0x0, 0x3E8]。
enIrisFNO	手动光圈大小, 根据光圈 F 值进行区分, 仅支持 P-Iris, 不支持 DC-Iris。

**【注意事项】**

- 对接 DC-Iris 镜头时, 若 ISP\_IRIS\_STATUS\_E 设置为 ISP\_IRIS\_KEEP, 手动光圈使能, u32HoldValue 可用于 DC-Iris 的调试, 此时 PWM 的占空比即为 u32HoldValue。
- 对接 P-Iris 镜头时, 若 ISP\_IRIS\_STATUS\_E 设置为 ISP\_IRIS\_KEEP, 手动光圈使能, enIrisFNO 可用于 P-Iris 的调试, 此时会控制 P-Iris 步进电机走到光圈 F 值与 enIrisFNO

最接近的位置。自动曝光模式下，P-Iris 手动光圈不生效，此时若要固定光圈为某个 F 值，可以将 enMaxIrisFNOTarget/ enMinIrisFNOTarget 设置为相同值来实现。

**【相关数据类型及接口】**

无。

#### 4.5.3.5 ISP\_DCIRIS\_ATTR\_S

**【说明】**

定义 DC-Iris 算法属性。

**【定义】**

```
typedef struct _ISP_DCIRIS_ATTR_S
{
    CVI_S32 s32Kp;
    CVI_S32 s32Ki;
    CVI_S32 s32Kd;CVI_U32 u32MinPwmDuty;CVI_U32 u32MaxPwmDuty;CVI_U32
    ↪u32OpenPwmDuty;
} ISP_DCIRIS_ATTR_S;
```

**【成员】**

成员名称	描述
s32Kp	比例增益，用于调节光圈的开关速度，该值越大光圈打开和关闭的速度越快。 该值过小光线剧烈变化收敛过程中容易出现振荡，该值过大容易出现超调，也会导致振荡。 该值的合理设置与电路特性和镜头相关。建议值为 7000。 取值范围为 [0, 100000]。
s32Ki	积分增益，用于调节光圈的开关速度，该值越大光圈打开和关闭的速度越快。 该值较小时，碰到强光收敛后画面会稳定在一个比较低的亮度上；该值较大时可能会导致强光场景光圈无法关闭。 该值的合理设置与电路特性和镜头相关。建议值为 100，该值一般不需要修改。 取值范围为 [0, 1000]。
s32Kd	微分增益，用于限制光线剧烈变化时光圈的开关速度，该值越大光线剧烈变化时光圈打开和关闭的速度越慢。 该值过大对于瞬间变化的亮度过于敏感，会导致场景亮度快速变化时画面出现振荡。 该值的合理设置与电路特性和镜头相关。建议值为 3000。 取值范围为 [0, 100000]。
u32MinPwmDuty	最小 PWM 占空比。该值越小过曝时光圈关闭速度越快，但容易导致光圈来回震荡。 该值的合理设置与电路特性和镜头相关。建议值为 250。 取值范围为 [0, 1000]。
u32MaxPwmDuty	最大 PWM 占空比。该值越大画面全黑时光圈打开速度越快，该值过小则可能导致退出光圈控制区域时光圈仍未达到最大，造成画面噪声严重。 该值的合理设置与电路特性和镜头相关。建议值为 950。 取值范围为 [0, 1000]。
u32OpenPwmDuty	光圈打开时的 PWM 占空比。 当画面亮度稳定并且 PWM 占空比大于该值一段时间后，退出光圈控制区域。 所以该值不能太小，否则容易导致光圈未达到最大就退出了光圈控制区域，造成画面噪声严重。 该值的合理设置与电路特性和镜头相关。建议值为 800。 取值范围为 [0, 1000]。

**【注意事项】**

- 当光圈关闭出现震荡时，一般意味着光圈关闭速度太快了，可以通过适当减小 s32Kp 和增大 u32MinPwmDuty 来解决。
- u32OpenPwmDuty 的取值要求在 u32MinPwmDuty 和 u32MaxPwmDuty 之间，必须确保该值能将光圈较快的打开。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDcirisAttr
- CVI\_ISP\_GetDcirisAttr

#### 4.5.3.6 ISP\_PIRIS\_ATTR\_S

**【说明】**

定义 P-Iris 属性。

**【定义】**

无。

**【成员】**

成员名称	描述
无	无

**【注意事项】**

目前尚未实作支持 P- Iris

**【相关数据类型及接口】**

无。

#### 4.5.3.7 ISP\_IRIS\_ATTR\_S

**【说明】**

定义 ISP 光圈属性。

**【定义】**

```
typedef struct _ISP_IRIS_ATTR_S {  
    CVI_BOOL bEnable;  
    ISP_OP_TYPE_E enOpType;  
    ISP_IRIS_TYPE_E enIrisType;  
    ISP_IRIS_STATUS_E enIrisStatus;  
    ISP_MI_ATTR_S stMIAttr;  
} ISP_IRIS_ATTR_S;
```

**【成员】**

成员名称	描述
bEnable	自动光圈使能。
enOpType	自动光圈或手动光圈模式选择。
enIrisType	光圈类型，DC-Iris 或 P-Iris。
enIrisStatus	光圈状态。
stMIAttr	手动光圈属性设置结构体。

**【注意事项】**

该值设置为 ISP\_IRIS\_OPEN 或 ISP\_IRIS\_CLOSE 时，光圈处于全开或全关的状态，可用于测试光圈控制电路和驱动是否正确。OPEN 和 CLOSE 的优先级高于光圈控制使能和手动/自动模式，光圈控制算法运行时，为保证其能够正常工作，需将该值设置为 ISP\_IRIS\_KEEP。

**【相关数据类型及接口】**

- CVI\_ISP\_SetIrisAttr
- CVI\_ISP\_GetIrisAttr

# 5 AWB

---

## 5.1 概述

色温随可见光的光谱成分变化而变化，在低色温光源下，白色物体偏红，在高色温光源下，白色物体偏蓝。人眼可根据大脑的记忆判断，识别物体的真实颜色，AWB 算法功能是降低外界光源对物体真实颜色的影响，使得我们采集的颜色信息转变为在理想日光光源下的无偏色信息

## 5.2 重要概念

- 色温：色温是按绝对黑体来定义的，光源的辐射在可见区和绝对黑体的辐射完全相同时，此时黑体的温度就称此光源的色温。
- 白平衡：在不同色温的光源下，白色在传感器中的响应会偏蓝或偏红。白平衡算法通过调整 R, G, B 三个颜色通道的强度，使白色真实呈现。

## 5.3 功能描述

AWB 模块有硬件的 WB 信息统计模块及 AWB 策略控制算法 firmware 两部分组成。

ISP 的 WB 信息统计模块判断 sensor 输出的每个像素是否满足用户设定的白点条件，计算所有满足条件的像素的 R、G、B 三个颜色通道平均值。

其支持将图像分成 M\*N (M 行 N 列) 区域，统计每个区域的 R、G、B 均值以及参与统计的白点个数，也支持输出整幅图像的 R、G、B 均值以及参与统计的白点个数。

AWB 工作原理如图 4-1 所示



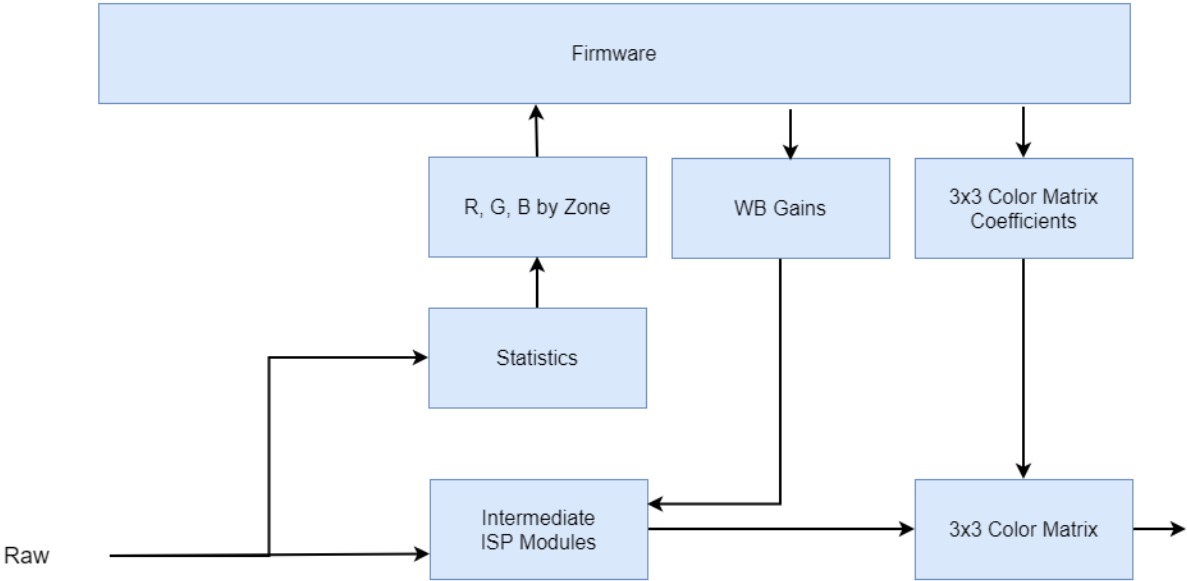


图 5.1: 图 4-1 AWB 工作原理图

## 5.4 API 参考

### 5.4.1 AWB 库接口

所有 AWB 库接口都只是针对 CVI AWB 库，如果客户自己实现 AWB 库，不需要管这些接口，同时也无法使用这些接口。

- `CVI_AWB_Register`：向 ISP 注册 AWB 库。
- `CVI_AWB_UnRegister`：向 ISP 注销 AWB 库。
- `CVI_AWB_SensorRegCallBack`：AWB 库提供的 sensor 注册的回调接口
- `CVI_AWB_SensorUnRegCallBack`：AWB 库提供的 sensor 注销的回调接口

#### 5.4.1.1 CVI\_AWB\_Register

**【描述】**

向 ISP 注册 AWB 库。

**【语法】**

```
CVI_S32 CVI_AWB_Register(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAwbLib	AWB 算法库结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: `cvi_awb.h`
- 库文件: `libawb.a`

**【注意】**

- 该接口调用了 ISP 库提供的 AWB 注册回调接口 `CVI_ISP_AWBLibRegCallBack` , 以实现 AWB 向 ISP 库注册的功能。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

#### 5.4.1.2 CVI\_AWB\_UnRegister

**【描述】**

向 ISP 反注册 AWB 库。

**【语法】**

```
CVI_S32 CVI_AWB_UnRegister(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAwbLib	AWB 算法库结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: `cvi_awb.h`

- 库文件: libawb.a

**【注意】**

- 该接口调用了 ISP 库提供的 AWB 反注册回调接口 CVI\_ISP\_AWBLibUnRegCallBack , 以实现 AWB 向 ISP 库反注册的功能。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

### 5.4.1.3 CVI\_AWB\_SensorRegCallBack

**【描述】**

AWB 库提供的 sensor 注册的回调接口。

**【语法】**

```
CVI_S32 CVI_AWB_SensorRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib, ISP_SNS_
↪ATTR_INFO_S *pstSnsAttrInfo, AWB_SENSOR_REGISTER_S *pstRegister);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAeLib	AWB 算法库结构体指针	输入
pstSnsAttrInfo	向 AWB 注册的 Sensor 的属性	输入
pstRegister	Sensor 注册结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件: cvi\_awb.h
- 库文件: libawb.a

**【注意】**

- SensorId 是 sensor 库中自定义的值, 主要用于校对向 ISP 注册的 sensor 和向 3A 注册的 sensor 是否为同一个 sensor。
- AWB 通过 sensor 注册的一系列回调接口, 获取差异化的初始化参数, 并控制 sensor。

- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

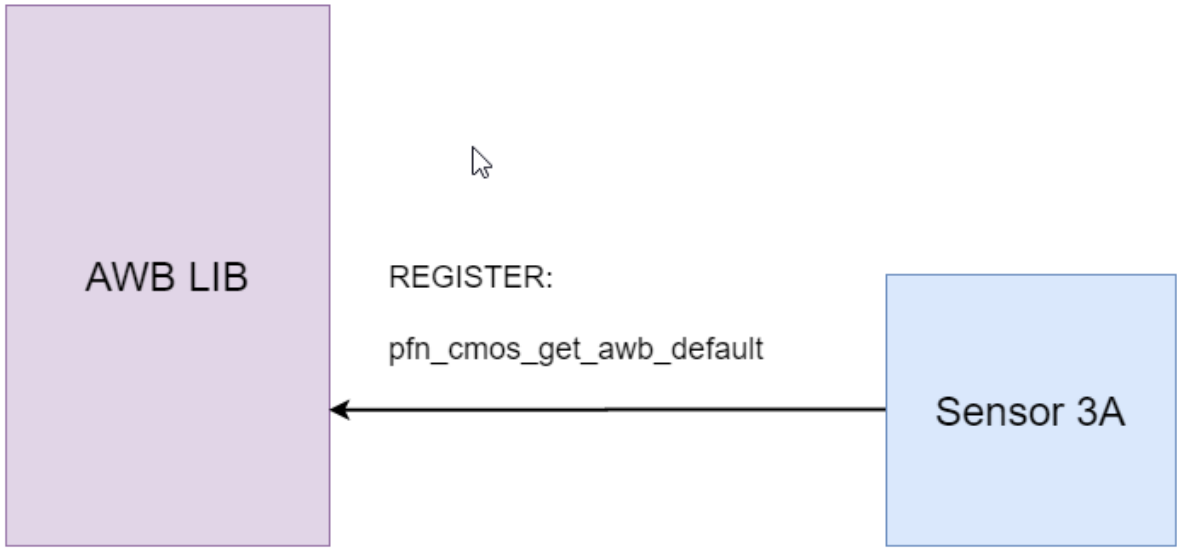


图 5.2: 图 4-2 AWB 库与 sensor 库间的接口

【举例】

无。

【相关主题】

无。

5.4.1.4 CVI\_AWB\_SensorUnRegCallBack

【描述】

AWB 库提供的 sensor 反注册的回调接口。

【语法】

```
CVI_S32 CVI_AWB_SensorUnRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib, SENSOR_
↪ID SensorId);
```

【参数】

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAwbLib	AWB 算法库结构体指针	输入
SensorId	向 AWB 反注册的 Sensor 的 Id	输入

【返回值】

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

- SensorId 是 sensor 库中自定义的值，主要用于校对向 ISP 注册的 sensor 和向 3A 注册的 sensor 是否为同一个 sensor。
- 此接口不支持多进程操作。
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

## 5.4.2 AWB 控制模块

- CVI\_ISP\_SetWBAttr：设置白平衡属性。
- CVI\_ISP\_GetWBAttr：获取白平衡属性。
- CVI\_ISP\_SetAWBAttrEx：设置白平衡扩展属性。
- CVI\_ISP\_GetAWBAttrEx：获取白平衡扩展属性。
- CVI\_ISP\_QueryWBInfo：获取当前白平衡增益系数，检测色温，饱和度值，颜色校正矩阵系数。
- CVI\_ISP\_SetAWBLogPath：使用 CVI Awb lib 时，存放 AWB 调试日志的路径。
- CVI\_ISP\_SetAWBLogName：使用 CVI Awb lib 时，存放 AWB 调试日志的名称。
- CVI\_ISP\_GetGrayWorldAwbInfo：获取灰世界 WB 信息。

### 5.4.2.1 CVI\_ISP\_SetWBAttr

**【描述】**

设置白平衡属性。

**【语法】**

```
CVI_S32 CVI_ISP_SetWBAttr(VI_PIPE ViPipe, const ISP_WB_ATTR_S *pstWBAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstWBAttr	白平衡属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

- 白平衡控制类型为自动时，AWB 算法自动调节白平衡系数。
- 白平衡控制类型为手动时，AWB 算法失效，需自行设定 Rgain、Ggain、Bgain。
- 环境色温、照度会影响白点的分布范围，CVI AWB 算法在运行过程中，会根据环境参数自动刷新 AWB Byaer 域统计信息的白点参数。用户希望在 CVI AWB 算法运行时修改 AWB 统计参数，需要调用 CVI\_ISP\_SetWBAttr 接口，关闭统计参数自动刷新功能。用户关闭统计参数自动刷新功能，到 ISP 收到配置并响应，有 2 帧的延时。因此，用户设置后需要等待 2 帧时间再修改 AWB 统计参数。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_GetWBAttr

**5.4.2.2 CVI\_ISP\_GetWBAttr****【描述】**

获取白平衡属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetWBAttr(VI_PIPE ViPipe, ISP_WB_ATTR_S *pstWBAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstWBAttr	白平衡属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetWBAttr

## 5.4.2.3 CVI\_ISP\_SetAWBAttrEx

**【描述】**

设置白平衡扩展属性。

**【语法】**

```
CVI_S32 CVI_ISP_SetAWBAttrEx(VI_PIPE ViPipe, const ISP_AWB_ATTR_EX_S_
↪ *pstAWBAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
PstAWBAttrEx	扩展白平衡属性结构体指针	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

- 当 CVI\_ISP\_SetWBAttr 接口成员 pstWBAttr->enAlgType 为 AWB\_ALG\_ADVANCE 时，此接口才有效。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_GetAWBAttrEx

**5.4.2.4 CVI\_ISP\_GetAWBAttrEx****【描述】**

获取白平衡扩展属性。

**【语法】**

```
CVI_S32 CVI_ISP_GetAWBAttrEx(VI_PIPE ViPipe, ISP_AWB_ATTR_EX_S *pstAWBAttrEx);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pstAWBAttrEx	扩返回展白平衡属性结构体指针	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetAWBAttrEx



#### 5.4.2.5 CVI\_ISP\_QueryWBInfo

**【描述】**

获取当前白平衡增益系数、检测色温、饱和度值、颜色校正矩阵系数。

**【语法】**

```
CVI_S32 CVI_ISP_QueryWBInfo(VI_PIPE ViPipe, ISP_WB_INFO_S *pstWBInfo);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
PstWBInfo	颜色相关状态参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

#### 5.4.2.6 CVI\_ISP\_SetAWBLogPath

**【描述】**

使用 CVI Awb lib 时，存放 AWB 调试日志的路径。

**【语法】**

```
CVI_S32 CVI_ISP_SetAWBLogPath(const char *szPath)
```

**【参数】**

参数名称	描述	输入/输出
szPath	调试日志路径	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

- 默认路径为 /var/log
- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

#### 5.4.2.7 CVI\_ISP\_SetAWBLogName

**【描述】**

使用 CVI Awb lib 时, 存放 AWB 调试日志的档名。

**【语法】**

```
CVI_S32 CVI_ISP_SetAWBLogName(const char *szName)
```

**【参数】**

参数名称	描述	输入/输出
szName	文件名	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见 <a href="#">错误码</a> 。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

- 预设文件名为 AwbLog0.txt

- 此接口在双系统 SDK 的 linux 侧暂不支持。

**【举例】**

无。

**【相关主题】**

无。

#### 5.4.2.8 CVI\_ISP\_GetGrayWorldAwbInfo

**【描述】**

获取灰度世界 WB 信息。

**【语法】**

```
CVI_S32 CVI_ISP_GetGrayWorldAwbInfo(VI_PIPE ViPipe, CVI_U16 *pRgain, CVI_U16 *pBgain);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	ViPipe 号	输入
pRgain	R 通道增益	输出
pBgain	B 通道增益	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，请参见错误码。

**【需求】**

- 头文件：cvi\_awb.h
- 库文件：libawb.a

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

## 5.5 数据类型

### 5.5.1 Register

- AWB\_SENSOR\_REGISTER\_S：定义 sensor 注册结构体。
- AWB\_SENSOR\_EXP\_FUNC\_S：定义 sensor 回调函数结构体。
- AWB\_SENSOR\_DEFAULT\_S：定义 AWB 算法库的初始化参数结构体。
- AWB\_SPEC\_SENSOR\_DEFAULT\_S：定义 SPEC AWB 算法的初始化参数结构体。
- AWB\_AGC\_TABLE\_S：定义饱和度初始化参数结构体。
- AWB\_CCM\_TAB\_S：定义不同色温下自动颜色校正矩阵系数。
- AWB\_CCM\_S：定义 CCM 颜色校正矩阵属性。

### 5.5.2 WB

- ISP\_AWB\_ATTR\_S：定义 ISP 自动白平衡属性。
- ISP\_AWB\_CBCR\_TRACK\_ATTR\_S：定义 Bayer 域统计信息的联动参数。
- ISP\_AWB\_LUM\_HISTGRAM\_ATTR\_S：定义白平衡的亮度直方图统计参数。
- ISP\_AWB\_ALG\_E：定义白平衡的计算方式属性。
- ISP\_AWB\_CT\_LIMIT\_ATTR\_S：定义白平衡的增益范围限制属性。
- ISP\_MWB\_ATTR\_S：定义 ISP 手动白平衡属性。
- ISP\_WB\_ATTR\_S：定义白平衡属性。
- ISP\_AWB\_ALG\_TYPE\_E：定义 AWB 算法属性。
- ISP\_AWB\_ATTR\_EX\_S：定义自动白平衡扩展属性。
- ISP\_AWB\_EXTRA\_LIGHTSOURCE\_INFO\_S：定义独立光源点的信息。
- ISP\_AWB\_IN\_OUT\_ATTR\_S：对场景作室内外的判定。
- ISP\_AWB\_MULTI\_LS\_TYPE\_E：定义混合光源下的 AWB 策略类型。
- ISP\_AWB\_INDOOR\_OUTDOOR\_STATUS\_E：定义 AWB 室内室外状态。
- ISP\_WB\_INFO\_S：定义白平衡、饱和度、颜色校正信息。

### 5.5.2.1 ISP\_MWB\_ATTR\_S

**【说明】**

定义手动 WB

**【定义】**

```
typedef struct _ISP_MWB_ATTR_S {  
    CVI_U16 u16Rgain;  
    CVI_U16 u16Grgain;  
    CVI_U16 u16Gbgain;  
    CVI_U16 u16Bgain;  
} ISP_MWB_ATTR_S;
```

**【成员】**

成员名称	描述
u16Rgain	MWB 时的 R 通道增益
u16Grgain	MWB 时的 Gr 通道增益
u16Gbgain	MWB 时的 Gb 通道增益
u16Bgain	MWB 时的 B 通道增益

**【注意事项】**

- RGB 通道增益一倍为 0x400

**【相关数据类型及接口】**

无。

### 5.5.2.2 ISP\_AWB\_CT\_LIMIT\_ATTR\_S

**【说明】**

定义白平衡的增益范围限制属性

**【定义】**

```
typedef struct _ISP_AWB_CT_LIMIT_ATTR_S {  
    CVI_BOOL bEnable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U16 u16HighRgLimit;  
    CVI_U16 u16HighBgLimit;  
    CVI_U16 u16LowRgLimit;  
    CVI_U16 u16LowBgLimit;  
} ISP_AWB_CT_LIMIT_ATTR_S;
```

**【成员】**

成员名称	描述
bEnable	白平衡的增益范围限制开关
enOpType	自动或手动设定自平衡的增益范围
u16HighRgLimit	手动模式下高色温下的最大 R 增益
u16HighBgLimit	手动模式下高色温下的最小 B 增益
u16LowRgLimit	手动模式下低色温下的最小 R 增益
u16LowBgLimit	手动模式下低色温下的最大 B 增益

**【注意事项】**

- 自动模式下 AWB 算法会自己计算上下限增益值, 当切换为手动模式时, 会使用上述四个 R.B 增益的限制值
- 一倍增益为 0x400

**【相关数据类型及接口】**

无。

**5.5.2.3 ISP\_AWB\_ALG\_E****【说明】**

定义 AWB 算法类别

**【定义】**

```
typedef enum _ISP_AWB_ALG_E {
    ALG_AWB,
    ALG_AWB_SPEC,
    ALG_BUTT
} ISP_AWB_ALG_E;
```

**【成员】**

成员名称	描述
AWB	一般 AWB 算法返回
ALG_AWB_SPEC	基于 machine learning 的 AWB 算法, 目前不支持。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 5.5.2.4 ISP\_AWB\_ATTR\_S

#### 【说明】

定义自动 WB 属性

#### 【定义】

```
typedef struct _ISP_AWB_ATTR_S {  
    CVI_BOOL bEnable;  
    CVI_U16 u16RefColorTemp;  
    CVI_U16 au16StaticWB[ISP_BAYER_CHN_NUM];  
    CVI_S32 as32CurvePara[AWB_CURVE_PARA_NUM];  
    ISP_AWB_ALG_TYPE_E enAlgType;  
    CVI_U8 u8RGStrength;  
    CVI_U8 u8BGStrength;  
    CVI_U16 u16Speed;  
    CVI_U16 u16ZoneSel;  
    CVI_U16 u16HighColorTemp;  
    CVI_U16 u16LowColorTemp;  
    ISP_AWB_CT_LIMIT_ATTR_S stCTLimit;  
    CVI_BOOL bShiftLimitEn;  
    CVI_U16 u16ShiftLimit[AWB_CURVE_BOUND_NUM];  
    CVI_BOOL bGainNormEn;  
    CVI_BOOL bNaturalCastEn;  
    ISP_AWB_CBCR_TRACK_ATTR_S stCbCrTrack;  
    ISP_AWB_LUM_HISTGRAM_ATTR_S stLumaHist;  
    CVI_BOOL bAWBZoneWtEn;  
    CVI_U8 au8ZoneWt[AWB_ZONE_ORIG_ROW * AWB_ZONE_ORIG_COLUMN];  
} ISP_AWB_ATTR_S;
```

#### 【成员】

成员名称	描述
bEnable	自动白平衡算法开关
u16RefColorTemp	静态白平衡的光源色温值, 建议为 5000K 灯源
au16StaticWB	静态白平衡系数, 由 AWB 标定工具给出。返回取值范围: [0, 0x3FFF]。
as32CurvePara	CurvePara[0-2] 普朗克曲线系数, 由 AWB 标定工具给出。普朗克曲线描绘白色块返回在不同色温的标准光源下的颜色表现。CurvePara[3-5] 色温曲线系数, 由 AWB 标定工具给出。色温曲线描绘白色块的颜色表现与色温的对应关系。
enAlgType	AWB 算法类别选择, AWB_ALG_LOWCOST, AWB_ALG_ADVANCE 两种 ISP_AWB_ATTR_EX_S 扩展属性只有在 AWB_ALG_ADVANCE 才有作用
u8RGStrength	自动白平衡 R 通道校准强度。 取值范围: [0, 255]
u8BGStrength	自动白平衡 B 通道校准强度。 取值范围: [0, 255]
u16Speed	自动白平衡算法收敛速度。 取值范围: [0, 4095]
u16ZoneSel	参数为 0 或 255 时, 采用近似灰世界的白平衡算法, 其他值则为进行分类筛选, 提升精度
u16HighColorTemp	自动白平衡算法的色温上限。 取值范围: [8000, 10000]
u16LowColorTemp	自动白平衡算法的色温下限。 取值范围: [0x0, u8HighColorTemp)
stCTLimit	手动或自动限制白平衡增益值
bShiftLimitEn	AWB 超过类白点范围的增益映像回白点范围的开关
u16ShiftLimit	AWB 计算白点范围参数
bGainNormEn	对 RGB 通道增益进行返回限制, 可以改善低色温、低照度场景的信噪比, 预设开启
bNaturalCastEn	低色温下 AWB 风格喜好开关。预设关闭
stCbCrTrack	AWB 统计范围与 ISO 的连动参数
stLumaHist	AWB 亮度与权重参数
bAWBZoneWtEn	画面分区权重开关, 预设关闭
au8ZoneWt	32x32 画面权重, 取值范围 [0, 0xFF] (尚未实作) 返回

## 【注意事项】

无。

## 【相关数据类型及接口】

无。



### 5.5.2.5 ISP\_AWB\_ALG\_TYPE\_E

**【说明】**

定义 AWB 算法属性

**【定义】**

```
typedef enum ISP_AWB_ALG_TYPE_E {  
    AWB_ALG_LOWCOST,  
    AWB_ALG_ADVANCE,  
    AWB_ALG_BUTT  
} ISP_AWB_ALG_TYPE_E;
```

**【成员】**

成员名称	描述
AWB_ALG_LOWCOST	简易低运算量的 AWB 算法
AWB_ALG_ADVANCE	进阶扩展 AWB 算法, 与 ISP_AWB_ATTR_EX_S 相关
AWB_ALG_BUTT	无效设定值返回

**【注意事项】**

无。

**【相关数据类型及接口】**

ISP\_AWB\_ATTR\_EX\_S

### 5.5.2.6 ISP\_AWB\_CBCR\_TRACK\_ATTR\_S

**【说明】**

定义 AWB 统计范围与 ISO 的连动参数

**【定义】**

```
typedef struct ISP_AWB_CBCR_TRACK_ATTR_S {  
    CVI_BOOL bEnable;  
    CVI_U16 au16CrMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 au16CrMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 au16CbMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 au16CbMin[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_AWB_CBCR_TRACK_ATTR_S;
```

**【成员】**

成员名称	描述
bEnable	AWB 统计范围与 ISO 的连动开关
au16CrMax	不同 ISO 下 R/G 的最大值
au16CrMin	不同 ISO 下 R/G 的最小值
au16CbMax	不同 ISO 下 B/G 的最大值
au16CbMin	不同 ISO 下 B/G 的最小值

**【注意事项】**

- 建议在低色温下标定 CrMax(R/G),CbMin(B/G)

**【相关数据类型及接口】**

无。

**5.5.2.7 ISP\_AWB\_LUM\_HISTGRAM\_ATTR\_S****【说明】**

定义 AWB 亮度与权重参数

**【定义】**

```
typedef struct _ISP_AWB_LUM_HISTGRAM_ATTR_S {  
    CVI_BOOL bEnable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 au8HistThresh[AWB_LUM_HIST_NUM];  
    CVI_U16 au16HistWt[AWB_LUM_HIST_NUM];  
} ISP_AWB_LUM_HISTGRAM_ATTR_S;
```

宏定义如下

```
#define AWB_LUM_HIST_NUM (6)
```

**【成员】**

成员名称	描述
bEnable	不同亮度是否开启权重, 预设开启
enOpType	自动模式:AWB 自动分配权重 手动模式: 用户可自行设定亮度分类与权重
au8HistThresh	亮度分类的阈值 (手动模式下有效)
au16HistWt	亮度分类的权重 (手动模式下有效)

**【注意事项】**

- au8HistThresh[0] 固定为 0,au8HistThresh[5] 固定为 255.
- au8HistThresh[i+1] 必须大于 au8HistThresh[i];
- au16HistWt 权重范围为 32~512

**【相关数据类型及接口】**

无。

## 5.5.2.8 ISP\_WB\_ATTR\_S

## 【说明】

定义白平衡属性

## 【定义】

```
typedef struct _ISP_WB_ATTR_S {
    CVI_BOOL bByPass;
    CVI_U8 u8AWBRunInterval;
    ISP_OP_TYPE_E enOpType;
    ISP_MWB_ATTR_S stManual;
    ISP_AWB_ATTR_S stAuto;
    ISP_AWB_ALG_E enAlgType;
    CVI_U8 u8DebugMode;
} ISP_WB_ATTR_S;
```

## 【成员】

成员名称	描述
bByPass	白平衡模块 Bypass 使能，默认值 CVI_FALSE。返回
u8AWBRunInterval	白平衡模块工作频率。 取值范围：[0x1, 0xFF]
enOpType	自动白平衡和手动白平衡切换。
stManual	手动参数
stAuto	自动参数
enAlgType	AWB 算法类别返回
u8DebugMode	Debug 时使用，一般不需设定

## 【注意事项】

- bByPass 为 TRUE 时, WB 其他参数设定不生效, RGB 通道增益固定为一倍 0x400
- u8AWBRunInterval 默认值为 6, 表示每 6 帧执行一次 AWB, 可根据需求增加此数值, 让 AWB 运行频率降低

## 【相关数据类型及接口】

无。

## 5.5.2.9 ISP\_AWB\_ATTR\_EX\_S

## 【说明】

定义自动白平衡扩展属性

## 【定义】

```
typedef struct _ISP_AWB_ATTR_EX_S {
    CVI_U8 u8Tolerance;
    CVI_U8 u8ZoneRadius;
    CVI_U16 u16CurveLLimit;
    CVI_U16 u16CurveRLimit;
```

(下页继续)

(续上页)

```

CVI_BOOL bExtraLightEn;
ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S stLightInfo[AWB_LS_NUM];
ISP_AWB_IN_OUT_ATTR_S stInOrOut;
CVI_BOOL bMultiLightSourceEn;
ISP_AWB_MULTI_LS_TYPE_E enMultiLSType;
CVI_U16 u16MultiLSScaler;
CVI_U16 au16MultiCTBin[AWB_CT_BIN_NUM];
CVI_U16 au16MultiCTWt[AWB_CT_BIN_NUM];
CVI_BOOL bFineTunEn;
CVI_U8 u8FineTunStrength;
//AWB Algo 6
struct ST_ISP_AWB_INTERFERENCE_S stInterference;
struct ST_ISP_AWB_SKIN_S stSkin;
struct ST_ISP_AWB_SKY_S stSky;
struct ST_ISP_AWB_GRASS_S stGrass;
struct ST_ISP_AWB_CT_WGT_S stCtLv;
struct ST_ISP_AWB_SHIFT_LV_S stShiftLv;
struct ST_ISP_AWB_REGION_S stRegion;
CVI_U8 adjBgainMode;
CVI_U8 reserve[239];
} ISP_AWB_ATTR_EX_S;

// 宏定义如下:
#define AWB_CT_BIN_NUM (8)
#define AWB_LS_NUM (4)

```

## 【成员】

成员名称	描述
u8Tolerance	AWB 调整的偏差返回范围，误差在此范围内时,AWB 不做调整
u8ZoneRadius	AWB 统计值分区的大小。该值越小,AWB 精度越高, 但会降低 AWB 算法稳定性
u16CurveLLimit	AWB 色温曲线返回的左边界限 (R/G,B/G), 取值范围: [0x0, 0x200]
u16CurveRLimit	AWB 色温曲线的右边界限 (R/G,B/G), 范围: [0x200, 0x3FF]
bExtraLightEn	是否开启独立光源
stLightInfo	AWB 计算时是否考虑色温曲线外的独立光源点, 最多四个独立点
stInOrOut	AWB 对场景做室内外判断的参数
bMultiLightSourceEn	AWB 检测当前场返回景是否为混合光源, 来调整饱和度或 CCM
enMultiLSType	调整饱和度或是 CCM
u16MultiLSScaler	当混合光源下调整饱和度或 CCM 的强度
au16MultiCTBin	色温分段参数, 须为递增序列
au16MultiCTWt	色温分段权重, 范围: [0x0, 0x400]
bFineTunEn	AWB 特殊色检测开关, 例如肤色
u8FineTunStrength	肤色、蓝色等特殊色检测的强度

## 【注意事项】

- ISP\_AWB\_ATTR\_EX\_S 扩展属性只有在 pstWBAttr->enAlgType 为

AWB\_ALG\_ADVANCE 才有作用

- u8Tolerance 为 AWB 的灵敏度参数, 值为 0 时, AWB 参数会实时更新, 值比较大时, 环境色温微微变化时并不会更新 AWB 参数, 画面会轻微的偏色, 建议户外为 0, 室内设置为 2
- u16CurveLLimit 取值 <512, 主要为排除绿色物体, u16CurveRLimit 取值 >512, 为排除偏紫色区域
- bMultiLightSourceEn 开启后, AWB 会判断场景是否为混光源, 在这样的场景下会降低饱和度或是 CCM, 以减少偏色
- WDR 模式下会自动关闭混光源侦测功能

#### 【相关数据类型及接口】

无。

#### 5.5.2.10 ISP\_AWB\_EXTRA\_LIGHTSOURCE\_INFO\_S

##### 【说明】

定义独立光源点的参数

##### 【定义】

```
typedef struct _ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S {
    CVI_U16 u16WhiteRgain;
    CVI_U16 u16WhiteBgain;
    CVI_U16 u16ExpQuant;
    CVI_U8 u8LightStatus;
    CVI_U8 u8Radius;
} ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S;
```

##### 【成员】

成员名称	描述
u16WhiteRgain	特殊光源点的 R 通道增益
u16WhiteBgain	特殊光源点的 B 通道增益
u16ExpQuant	根据外在亮度做判断。 ExpQuant 为开启的亮度限制值例如 ExpQuant = 6, 表示 LV6 以下返回开启此 WB 光源点 (一般夜景为 LV6 以下) ExpQuant = 106 表示 LV6 以上开启 ExpQuant = 112 表示 LV12 以上开启 (LV12 一般为户外)
u8LightStatus	特殊光源点的种类, 0: 不作动 1: 加入光源点返回 2: 删除光源点附近的计算
u8Radius	特殊光源点的区域大小, 取值范围: [0x0, 0xFF]

##### 【注意事项】

无。

## 【相关数据类型及接口】

无。

## 5.5.2.11 ISP\_AWB\_IN\_OUT\_ATTR\_S

## 【说明】

定义 AWB 对场景做户外室内判断的参数

## 【定义】

```
typedef struct _ISP_AWB_IN_OUT_ATTR_S {
    CVI_BOOL bEnable;
    ISP_OP_TYPE_E enOpType;
    ISP_AWB_INDOOR_OUTDOOR_STATUS_E enOutdoorStatus;
    CVI_U32 u32OutThresh;
    CVI_U16 u16LowStart;
    CVI_U16 u16LowStop;
    CVI_U16 u16HighStart;
    CVI_U16 u16HighStop;
    CVI_BOOL bGreenEnhanceEn;
    CVI_U u8OutShiftLimit;
} ISP_AWB_IN_OUT_ATTR_S;
```

## 【成员】

成员名称	描述
bEnable	判断场景为室内室外的开关
enOpType	判断室内室外（自动或手动）
enOutdoorStatus	室内或室外模式（手动模式下）
u32OutThresh	判定室内室外的阈值，亮度小于时，则判定为室内，户外 LV 大多超过 15
u16LowStart	将低色温的权重返回重拉低，低色温区的起始点，建议为 5000K
u16LowStop	将低色温的权重返回拉低，低色温区的终止点，建议为 4500K，取值范围：(0,u16LowStart)
u16HighStart	将高色温的权重拉低，高色温区的起始点，建议为 6500K，取值范围：(u16LowStart, 0xFFFF]
u16HighStop	将高色温的权重返回拉低，高色温区的终止点，建议为 8000K，取值范围：(u16HighStart, 0xFFFF]
bGreenEnhanceEn	在绿色植物场景下，对绿色通道增加的开关
u8OutShiftLimit	当判定为户外场景时，AWB 算法的白点范围限制

## 【注意事项】

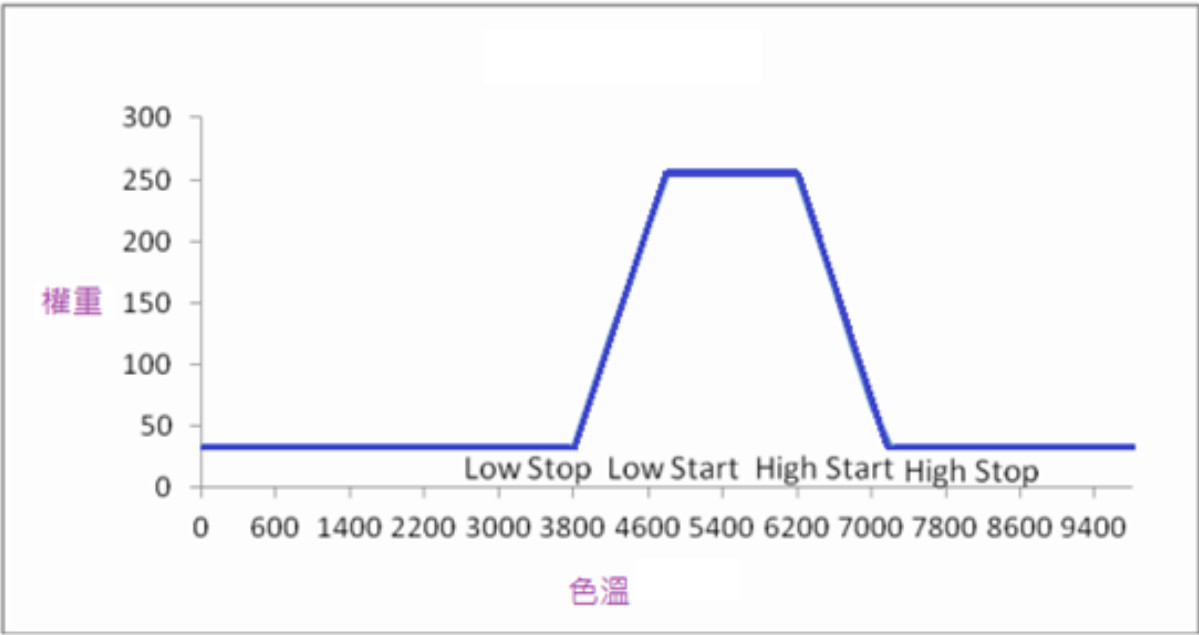
- 户外亮度大多 LV 大于 15 所以 u32OutThresh 建议为大于 15
- 四个参数的取值范围要求：u16LowStop < u16LowStart < u16HighStart < u16HighStop

## 【相关数据类型及接口】

如下图，u16LowStop 为 3800K，u16LowStart 为 5000K

u16HighStart 为 6200K，u16HighStop 为 7200K

一般权重为 32, 室外色温最高权重为 256



5.5.2.12 ISP\_AWB\_MULTI\_LS\_TYPE\_E

【说明】

定义混光源下 AWB 的策略

【定义】

```
typedef enum ISP_AWB_MULTI_LS_TYPE_E {
    AWB_MULTI_LS_SAT,
    AWB_MULTI_LS_CCM,
    AWB_MULTI_LS_BUTT
} ISP_AWB_MULTI_LS_TYPE_E;
```

【成员】

成员名称	描述
AWB_MULTI_LS_SAT	混合光源下调整饱和度
AWB_MULTI_LS_CCM	混合光源下调整 CCM
AWB_MULTI_LS_BUTT	无效

【注意事项】

无。

【相关数据类型及接口】

无。

### 5.5.2.13 ISP\_AWB\_INDOOR\_OUTDOOR\_STATUS\_E

**【说明】**

定义 AWB 室内室外状态

**【定义】**

```
typedef enum ISP_AWB_INDOOR_OUTDOOR_STATUS_E {  
    AWB_INDOOR_MODE,  
    AWB_OUTDOOR_MODE,  
    AWB_INDOOR_OUTDOOR_BUTT  
} ISP_AWB_INDOOR_OUTDOOR_STATUS_E;
```

**【成员】**

成员名称	描述
AWB_INDOOR_MODE	室内模式
AWB_OUTDOOR_MODE	室外模式
AWB_INDOOR_OUTDOOR_BUTT	无效

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 5.5.2.14 ISP\_WB\_INFO\_S

**【说明】**

定义白平衡，饱和度，颜色校正信息

**【定义】**

```
typedef struct ISP_WB_INFO_S {  
    CVI_U16 u16Rgain;  
    CVI_U16 u16Grgain;  
    CVI_U16 u16Gbgain;  
    CVI_U16 u16Bgain;  
    CVI_U16 u16Saturation;  
    CVI_U16 u16ColorTemp;  
    CVI_U16 au16CCM[CCM_MATRIX_SIZE];  
    CVI_U16 u16LS0CT;  
    CVI_U16 u16LS1CT;  
    CVI_U16 u16LS0Area;  
    CVI_U16 u16LS1Area;  
    CVI_U8 u8MultiDegree;  
    CVI_U16 u16ActiveShift;  
    CVI_U32 u32FirstStableTime;  
    ISP_AWB_INDOOR_OUTDOOR_STATUS_E enInOutStatus;  
    CVI_S16 s16Bv;  
} ISP_WB_INFO_S;
```



## 【成员】

成员名称	描述
u16Rgain	当前 R 通道增益值
u16Grgain	当前 Gr 通道增益值
u16Gbgain	当前 Gb 通道增益值
u16Bgain	当前 B 通道增益值
u16Saturation	目前饱和度返回
u16ColorTemp	目前推算的色温值
au16CCM	当前的颜色校正矩阵，10bit 小数精度。bit 15 是符号位，0 表示正数，1 表示负数，例如 0x8010 表示-16
u16LS0CT	混光源场景的主光源色温
u16LS1CT	混光源场景的次光源色温
u16LS0Area	混光源场景的主光源面积
u16LS1Area	混光源场景的次光源面积
u8MultiDegree	目前场景是混光源的机率
u16ActiveShift	目前白平衡范围的限制值
u32FirstStableTime	首次 AWB 收敛稳定的时间，以帧数为单位
ISP_AWB_INDOOR_OUTDOOR_RESULT	室内室外检测结果
s16Bv	当前环境的 bv 值返回

## 【注意事项】

无。

## 【相关数据类型及接口】

无。

# 6 IMP

---

# 7 BlackLevel

---

## 7.1 功能描述

Black Level 为当外界没有任何光线照入时，Sensor 仍然输出的亮度值。ISP 需要扣除，以使亮度、颜色正常。

## 7.2 API 参考

- CVI\_ISP\_SetBlackLevelAttr：设置黑电平属性参数
- CVI\_ISP\_GetBlackLevelAttr：获取黑电平属性参数

### 7.2.1 CVI\_ISP\_SetBlackLevelAttr

#### 【描述】

设置黑电平属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetBlackLevelAttr(VI_PIPE ViPipe, const ISP_BLACK_LEVEL_ATTR_S_
→ *pstBlackLevelAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstBlackLevelAttr	黑电平属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetBlackLevelAttr](#)

## 7.2.2 CVI\_ISP\_GetBlackLevelAttr

**【描述】**

获取黑电平属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetBlackLevelAttr(VI_PIPE ViPipe, ISP_BLACK_LEVEL_ATTR_S_
↪ *pstBlackLevelAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstBlackLevelAttr	黑电平属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetBlackLevelAttr](#)

## 7.3 数据类型

- `ISP_BLACK_LEVEL_MANUAL_ATTR_S`：定义黑电平手动属性
- `ISP_BLACK_LEVEL_AUTO_ATTR_S`：定义黑电平自动属性
- `ISP_BLACK_LEVEL_ATTR_S`：定义黑电平属性

### 7.3.1 `ISP_BLACK_LEVEL_MANUAL_ATTR_S`

**【说明】**

定义黑电平手动属性

**【定义】**

```
typedef struct _ISP_BLACK_LEVEL_MANUAL_ATTR_S {  
    CVI_U16 OffsetR;  
    CVI_U16 OffsetGr;  
    CVI_U16 OffsetGb;  
    CVI_U16 OffsetB;  
    CVI_U16 OffsetR2;  
    CVI_U16 OffsetGr2;  
    CVI_U16 OffsetGb2;  
    CVI_U16 OffsetB2;  
} ISP_BLACK_LEVEL_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
OffsetR	BLC R 像素暗电流值返回值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16 返回值
OffsetGr	BLC GR 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetGb	BLC GB 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetB	BLC B 像素暗电流值返回值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetR2	BLC 第二级 R 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16 返回值
OffsetGr2	BLC 第二级 GR 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetGb2	BLC 第二级 GB 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetB2	BLC 第二级 B 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetBlackLevelAttr
- CVI\_ISP\_GetBlackLevelAttr

### 7.3.2 ISP\_BLACK\_LEVEL\_AUTO\_ATTR\_S

**【说明】**

定义黑电平自动属性

**【定义】**

```
typedef struct _ISP_BLACK_LEVEL_AUTO_ATTR_S {
    CVI_U16 OffsetR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 OffsetGr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 OffsetGb[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 OffsetB[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 OffsetR2[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 OffsetGr2[ISP_AUTO_ISO_STRENGTH_NUM];
}
```

(下页继续)

(续上页)

```

CVI_U16 OffsetGb2[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U16 OffsetB2[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_BLACK_LEVEL_AUTO_ATTR_S;

```

**【成员】**

成员名称	描述
OffsetR	BLC R 像素暗电流值返回值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetGr	BLC GR 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetGb	BLC GB 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetB	BLC B 像素暗电流值返回值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetR2	BLC 第二级 R 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16 返回值
OffsetGr2	BLC 第二级 GR 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetGb2	BLC 第二级 GB 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
OffsetB2	BLC 第二级 B 像素暗电流值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetBlackLevelAttr
- CVI\_ISP\_GetBlackLevelAttr

### 7.3.3 ISP\_BLACK\_LEVEL\_ATTR\_S

**【说明】**

定义黑电平属性

**【定义】**

```
typedef struct _ISP_BLACK_LEVEL_ATTR_S {  
    CVI_U8 Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_BLACK_LEVEL_MANUAL_ATTR_S stManual;  
    ISP_BLACK_LEVEL_AUTO_ATTR_S stAuto;  
} ISP_BLACK_LEVEL_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	BLC 模块使能 取值范围：[0x0, 0x1] 数据类型：CVI_U8
enOpType	选择手动或自动模式返回值
UpdateInterval	影响参数更新间隔，返回值越大画面变化越慢，效能越好。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
stManual	手动参数
stAuto	自动参数

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetBlackLevelAttr
- CVI\_ISP\_GetBlackLevelAttr



# 8 DPC

---

## 8.1 功能描述

DPC 全称为 Defect Pixel Correction，可以修正 Sensor 上有问题的坏点。

修正方法有两种，分别为静态校正与动态校正。

- 静态校正：校正时分为暗点校正与亮点校正。  
亮点校正时，把镜头遮黑，启动坏点标定程序。  
暗点校正时，不遮黑在平坦背景时如灰度箱前，调整曝光让图像整体亮度约为 50%。  
静态坏点最多允许 4095 个坏点。
- 动态校正：使用此方法时，不使用校正数据而是直接动态判断坏点，并且加以修正。在低噪时对画面偏色有帮助，但若强度太强可能会致画面细节降低。

## 8.2 API 参考

- CVI\_ISP\_SetDPDynamicAttr：设置动态坏点校正属性
- CVI\_ISP\_GetDPDynamicAttr：获取动态坏点校正属性
- CVI\_ISP\_SetDPCalibrate：设置静态坏点标定参数
- CVI\_ISP\_GetDPCalibrate：获取静态坏点标定参数
- CVI\_ISP\_SetDPStaticAttr：设置静态坏点校正属性
- CVI\_ISP\_GetDPStaticAttr：获取静态坏点校正属性

## 8.2.1 CVI\_ISP\_SetDPDynamicAttr

### 【描述】

设置动态坏点校正属性

### 【语法】

```
CVI_S32 CVI_ISP_SetDPDynamicAttr(VI_PIPE ViPipe, const ISP_DP_DYNAMIC_ATTR_S_
↪ *pstDPCDynamicAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPCDynamicAttr	动态坏点校正属性	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无。

### 【举例】

无。

### 【相关主题】

- CVI\_ISP\_GetDPDynamicAttr

## 8.2.2 CVI\_ISP\_GetDPDynamicAttr

### 【描述】

获取动态坏点校正属性参数

### 【语法】

```
CVI_S32 CVI_ISP_GetDPDynamicAttr(VI_PIPE ViPipe, ISP_DP_DYNAMIC_ATTR_S_
↪ *pstDPCDynamicAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPDynamicAttr	静态坏点标定参数	输出

## 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

## 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

## 【注意】

无。

## 【举例】

无。

## 【相关主题】

- [CVI\\_ISP\\_SetDPDynamicAttr](#)

## 8.2.3 CVI\_ISP\_SetDPCalibrate

## 【描述】

设置静态坏点校正属性

## 【语法】

```
CVI_S32 CVI_ISP_SetDPCalibrate(VI_PIPE ViPipe, const ISP_DP_CALIB_ATTR_S_
↪ *pstDPCalibAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPCalibrateAttr	静态坏点校正属性	输入

## 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetDPCalibrate](#)

## 8.2.4 CVI\_ISP\_GetDPCalibrate

**【描述】**

获取静态坏点标定参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDPCalibrate(VI_PIPE ViPipe, ISP_DP_CALIB_ATTR_S_
↪*pstDPCalibAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPCalibrateAttr	静态坏点标定参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetDPCalibrate](#)

## 8.2.5 CVI\_ISP\_SetDPStaticAttr

### 【描述】

设置静态坏点校正属性

### 【语法】

```
CVI_S32 CVI_ISP_SetDPStaticAttr(VI_PIPE ViPipe, const ISP_DP_STATIC_ATTR_S_
↪*pstDPStaticAttr);
```

### 【参数】

参数名称描述	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPStaticAttr	静态坏点校正属性	输入

### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

### 【注意】

无。

### 【举例】

无。

### 【相关主题】

- [CVI\\_ISP\\_GetDPStaticAttr](#)

## 8.2.6 CVI\_ISP\_GetDPStaticAttr

### 【描述】

获取静态坏点校正属性

### 【语法】

```
CVI_S32 CVI_ISP_GetDPStaticAttr(VI_PIPE ViPipe, ISP_DP_STATIC_ATTR_S_
↪*pstDPStaticAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDPStaticAttr	静态坏点校正属性	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

`CVI_ISP_SetDPStaticAttr`

## 8.3 数据类型

- `ISP_DP_DYNAMIC_MANUAL_ATTR_S`: 动态坏点校正手动属性
- `ISP_DP_DYNAMIC_AUTO_ATTR_S`: 动态坏点校正自动属性
- `ISP_DP_DYNAMIC_ATTR_S`: 动态坏点校正属性
- `ISP_DP_CALIB_ATTR_S`: 静态坏点标定参数
- `ISP_DP_STATIC_ATTR_S`: 静态坏点校正属性

### 8.3.1 ISP\_DP\_DYNAMIC\_MANUAL\_ATTR\_S

**【说明】**

动态坏点校正手动属性

**【定义】**

```
typedef struct _ISP_DP_DYNAMIC_MANUAL_ATTR_S {
    CVI_U8 ClusterSize;
    CVI_U8 BrightDefectToNormalPixRatio;
    CVI_U8 DarkDefectToNormalPixRatio;
    CVI_U8 FlatThreR;
    CVI_U8 FlatThreG;
    CVI_U8 FlatThreB;
    CVI_U8 FlatThreMinG;
    CVI_U8 FlatThreMinRB;
} ISP_DP_DYNAMIC_MANUAL_ATTR_S;
```

### 【成员】

成员名称	描述
ClusterSize	群聚坏点面积上限，值越高越能修正群聚坏点，但可能会造成高频区域解像力的衰减 取值范围：[0x0, 0x3] 数据类型：CVI_U8
BrightDefectToNormalPixRatio	可视亮坏点值与周围像素的倍率 (Q4.4) 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkDefectToNormalPixRatio	可视暗坏点值与周围像素的倍率 (Q4.4) 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreR	R 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreG	G 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreB	B 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreMinG	G 通道判别平坦区最小临界值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreMinRB	RB 通道判别平坦区最小临界值 取值范围：[0x0, 0xff] 数据类型：CVI_U8

### 【注意事项】

无。

### 【相关数据类型及接口】

- [CVI\\_ISP\\_SetDPDynamicAttr](#)
- [CVI\\_ISP\\_GetDPDynamicAttr](#)

### 8.3.2 ISP\_DP\_DYNAMIC\_AUTO\_ATTR\_S

#### 【说明】

动态坏点校正自动属性

#### 【定义】

```
typedef struct _ISP_DP_DYNAMIC_AUTO_ATTR_S {
    CVI_U8 ClusterSize[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 BrightDefectToNormalPixRatio[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DarkDefectToNormalPixRatio[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 FlatThreR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 FlatThreG[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 FlatThreB[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 FlatThreMinG[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 FlatThreMinRB[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DP_DYNAMIC_AUTO_ATTR_S;
```

#### 【成员】

成员名称	描述
ClusterSize	群聚坏点面积上限，值越高越能修正群聚坏点，但可能会造成高频区域解像力的衰减 取值范围：[0x0, 0x3] 数据类型：CVI_U8
BrightDefectToNormalPixRatio	可视亮坏点值与周围像素的倍率 (Q4.4) 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkDefectToNormalPixRatio	可视暗坏点值与周围像素的倍率 (Q4.4) 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreR	R 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreG	G 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreB	B 通道判别平坦区临界值，值越小越能保留边缘信息 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreMinG	G 通道判别平坦区最小临界值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FlatThreMinRB	RB 通道判别平坦区最小临界值 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】



- CVI\_ISP\_SetDPDynamicAttr
- CVI\_ISP\_GetDPDynamicAttr

### 8.3.3 ISP\_DP\_DYNAMIC\_ATTR\_S

#### 【说明】

动态坏点校正属性

#### 【定义】

```
typedef struct _ISP_DP_DYNAMIC_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_U32 DynamicDPCEnable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_DP_DYNAMIC_MANUAL_ATTR_S stManual;  
    ISP_DP_DYNAMIC_AUTO_ATTR_S stAuto;  
} ISP_DP_DYNAMIC_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	DPC 模块使能 0: 关闭 1: 使能 取值范围: [0, 1] 数据类型: CVI_BOOL
DynamicDPCEnable	动态坏点校正功能使能 数据类型: CVI_U32
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stManual	手动模式下参数
stAuto	自动模式下参数

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- CVI\_ISP\_SetDPDynamicAttr
- CVI\_ISP\_GetDPDynamicAttr

### 8.3.4 ISP\_DP\_CALIB\_ATTR\_S

**【说明】**

静态坏点标定参数

**【定义】**

```
typedef struct _ISP_DP_CALIB_ATTR_S {  
    CVI_BOOL EnableDetect;  
    CVI_STATIC_DP_TYPE_E StaticDPType;  
    CVI_U8 StartThresh;  
    CVI_U16 CountMax;  
    CVI_U16 CountMin;  
    CVI_U16 TimeLimit;  
    CVI_BOOL saveFileEn;  
  
    // read only  
    CVI_U32 Table[STATIC_DP_COUNT_MAX];  
    CVI_U8 FinishThresh;  
    CVI_U16 Count;  
    ISP_STATUS_E Status;  
} ISP_DP_CALIB_ATTR_S;
```

**【成员】**

成员名称	描述
EnableDetect	静态坏点标定使能 取值范围：[0, 1] 数据类型：CVI_BOOL
StaticDPTType	静态坏点标定类型 取值范围：[0x0, 0x1] 数据类型：CVI_STATIC_DP_TYPE_E
StartThresh	静态坏点标定开始时的检测门限值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
CountMax	允许静态坏点的最大个数 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
CountMin	允许静态坏点的最小个数 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
TimeLimit	允许标定超时门限值 取值范围：[0x0, 0x640] 数据类型：CVI_U16
saveFileEn	是否保存 raw 图 取值范围：[0, 1] 数据类型：CVI_BOOL
Table[4096]	只读，亮暗坏点坐标值查找表，低 29bit 有效，[12:0]bit 为坏点的水平坐标，[28:16]bit 为坏点的垂直坐标 取值范围：[0x0, 0x1fff1fff] 数据类型：CVI_U32
FinishThresh	只读，静态坏点标定结束时的检测门限值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
Count	只读，标定出的静态坏点的个数 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
Status	只读，静态坏点标定结果状态信息 取值范围：[0x0, 0x2] 数据类型：ISP_STATUS_E

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDPCalibrate
- CVI\_ISP\_GetDPCalibrate

### 8.3.5 ISP\_DP\_STATIC\_ATTR\_S

**【说明】**

静态坏点校正属性

**【定义】**

```
typedef struct _ISP_DP_STATIC_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_U16 BrightCount;  
    CVI_U16 DarkCount;  
    CVI_U32 BrightTable[STATIC_DP_COUNT_MAX];  
    CVI_U32 DarkTable[STATIC_DP_COUNT_MAX];  
    CVI_BOOL Show; // not support yet  
} ISP_DP_STATIC_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	静态坏点 DPC 使能 0: 关闭 1: 使能 取值范围: [0, 1] 数据类型: CVI_BOOL
BrightCount	亮坏点个数 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
DarkCount	暗坏点个数 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
BrightTable[4095]	亮坏点坐标信息, 低 29bit 有效, [12:0]bit 为坏点水平坐标, [28:16]bit 为坏点垂直坐标。 取值范围: [0x0, 0x1fff1fff] 数据类型: CVI_U32
DarkTable[4095]	暗的坏点坐标值, 低 29bit 有效, [12:0]bit 为坏点水平坐标, [28:16]bit 为坏点垂直坐标。 取值范围: [0x0, 0x1fff1fff] 数据类型: CVI_U32
Show	静态坏点显示使能 not support yet 取值范围: [0, 1] 数据类型: CVI_BOOL

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDPStaticAttr
- CVI\_ISP\_GetDPStaticAttr

# 9 MeshShading

## 9.1 功能描述

以网格形式对镜头四周暗角进行矫正

## 9.2 API 参考

- `CVI_ISP_SetMeshShadingAttr`: 设置 Mesh Shading 算法参数
- `CVI_ISP_GetMeshShadingAttr`: 获取 Mesh Shading 算法参数
- `CVI_ISP_SetMeshShadingGainLutAttr`: 设置 LSC 网格形式补偿增益表
- `CVI_ISP_GetMeshShadingGainLutAttr`: 获取 LSC 网格形式补偿增益表

### 9.2.1 CVI\_ISP\_SetMeshShadingAttr

#### 【描述】

设置 Mesh Shading 算法参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetMeshShadingAttr(VI_PIPE ViPipe, const ISP_MESH_SHADING_ATTR_S_
→*pstMeshShadingAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMeshShadingAttr	Mesh Shading 算法参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetMeshShadingAttr](#)

## 9.2.2 CVI\_ISP\_GetMeshShadingAttr

**【描述】**

获取 Mesh Shading 算法参数

**【语法】**

```
CVI_S32 CVI_ISP_GetMeshShadingAttr(VI_PIPE ViPipe, ISP_MESH_SHADING_ATTR_S_
↪ *pstMeshShadingAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMeshShadingAttr	Mesh Shading 算法参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- `CVI_ISP_SetMeshShadingAttr`

### 9.2.3 CVI\_ISP\_SetMeshShadingGainLutAttr

#### 【描述】

设置 LSC 网格形式补偿增益表

#### 【语法】

```
CVI_S32 CVI_ISP_SetMeshShadingGainLutAttr(VI_PIPE ViPipe, const ISP_MESH_SHADING_
↪GAIN_LUT_ATTR_S *pstMeshShadingGainLutAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMeshShading-GainLutAttr	LSC 网格线形式补偿增益表	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

#### 【注意】

无。

#### 【举例】

无。

#### 【相关主题】

- `CVI_ISP_GetMeshShadingGainLutAttr`

## 9.2.4 CVI\_ISP\_GetMeshShadingGainLutAttr

### 【描述】

获取 LSC 网格形式补偿增益表

### 【语法】

```
CVI_S32 CVI_ISP_GetMeshShadingGainLutAttr(VI_PIPE ViPipe, ISP_MESH_SHADING_GAIN_
↳ LUT_ATTR_S *pstMeshShadingGainLutAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMeshShading-GainLutAttr	LSC 网格线形式补偿增益表	输出

### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无。

### 【举例】

无。

### 【相关主题】

- CVI\_ISP\_SetMeshShadingGainLutAttr

## 9.3 数据类型

- ISP\_MESH\_SHADING\_MANUAL\_ATTR\_S: Mesh Shading 算法手动参数
- ISP\_MESH\_SHADING\_AUTO\_ATTR\_S: Mesh Shading 算法自动参数
- ISP\_MESH\_SHADING\_ATTR\_S: Mesh Shading 算法参数
- ISP\_MESH\_SHADING\_GAIN\_LUT\_S: LSC 网格线形式补偿增益表细项
- ISP\_MESH\_SHADING\_GAIN\_LUT\_ATTR\_S: LSC 网格线形式补偿增益表



### 9.3.1 ISP\_MESH\_SHADING\_MANUAL\_ATTR\_S

**【说明】**

Mesh Shading 算法手动参数

**【定义】**

```
typedef struct _ISP_MESH_SHADING_MANUAL_ATTR_S {  
    CVI_U16 MeshStr;  
} ISP_MESH_SHADING_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
MeshStr	LSC 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetMeshShadingAttr
- CVI\_ISP\_GetMeshShadingAttr

### 9.3.2 ISP\_MESH\_SHADING\_AUTO\_ATTR\_S

**【说明】**

Mesh Shading 算法自动参数

**【定义】**

```
typedef struct _ISP_MESH_SHADING_AUTO_ATTR_S {  
    CVI_U16 MeshStr[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_MESH_SHADING_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
MeshStr	LSC 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetMeshShadingAttr
- CVI\_ISP\_GetMeshShadingAttr

### 9.3.3 ISP\_MESH\_SHADING\_ATTR\_S

#### 【说明】

Mesh Shading 算法参数

#### 【定义】

```
typedef struct _ISP_MESH_SHADING_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_BOOL OverflowProtection;  
    ISP_MESH_SHADING_MANUAL_ATTR_S stManual;  
    ISP_MESH_SHADING_AUTO_ATTR_S stAuto;  
} ISP_MESH_SHADING_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	LSC 功能使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	选择手动或自动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
OverflowProtection	避免过曝区色偏 取值范围：[0, 1] 数据类型：CVI_BOOL
stManual	手动参数
stAuto	自动参数

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- CVI\_ISP\_SetMeshShadingAttr
- CVI\_ISP\_GetMeshShadingAttr

### 9.3.4 ISP\_MESH\_SHADING\_GAIN\_LUT\_S

**【说明】**

网格线形式补偿增益表细项

**【定义】**

```
typedef struct _ISP_MESH_SHADING_GAIN_LUT_S {  
    CVI_U16 ColorTemperature;  
    CVI_U16 RGain[CVI_ISP_LSC_GRID_POINTS];  
    CVI_U16 GGain[CVI_ISP_LSC_GRID_POINTS];  
    CVI_U16 BGain[CVI_ISP_LSC_GRID_POINTS];  
} ISP_MESH_SHADING_GAIN_LUT_S;
```

**【成员】**

成员名称	描述
ColorTemperature	色温自适应 LSC 补偿增益表所对应之色温，单位为 K 取值范围：[0x0, 0x7530] 数据类型：CVI_U16
RGain	红色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
GGain	绿色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
BGain	蓝色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetMeshShadingGainLutAttr
- CVI\_ISP\_GetMeshShadingGainLutAttr

### 9.3.5 ISP\_MESH\_SHADING\_GAIN\_LUT\_ATTR\_S

**【说明】**

LSC 网格线形式补偿增益表

**【定义】**

```
typedef struct _ISP_MESH_SHADING_GAIN_LUT_ATTR_S {  
    CVI_U8 Size;
```

(下页继续)

(续上页)

```
ISP_MESH_SHADING_GAIN_LUT_S LscGainLut[ISP_MLSC_COLOR_TEMPERATURE_
↪SIZE];
} ISP_MESH_SHADING_GAIN_LUT_ATTR_S;
```

【成员】

成员名称	描述
Size	色温自适应 LSC 补偿增益表数量 取值范围：[0x1, 0x7] 数据类型：CVI_U8
LscGainLut	LSC 网格线形式补偿增益表

【注意事项】

无。

【相关数据类型及接口】

- CVI\_ISP\_SetMeshShadingGainLutAttr
- CVI\_ISP\_GetMeshShadingGainLutAttr

# 10 RadialShading

## 10.1 功能描述

以同心圆形式对镜头四周暗角进行矫正

## 10.2 API 参考

- `CVI_ISP_SetRadialShadingAttr` : 设置 Radial LSC 参数
- `CVI_ISP_GetRadialShadingAttr` : 获取 Radial LSC 参数
- `CVI_ISP_SetRadialShadingGainLutAttr` : 设置 LSC Radius 形式补偿增益表
- `CVI_ISP_GetRadialShadingGainLutAttr` : 获取 LSC Radius 形式补偿增益表

### 10.2.1 CVI\_ISP\_SetRadialShadingAttr

#### 【描述】

设置 Radial LSC 参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetRadialShadingAttr(VI_PIPE ViPipe, const ISP_RADIAL_SHADING_
↪ATTR_S *pstRadialShadingAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRadialShadingAttr	设置 Radial LSC 参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

cv180x & cv181x 不支持此功能

**【举例】**

无

**【相关主题】**

- [CVI\\_ISP\\_GetRadialShadingAttr](#)

## 10.2.2 CVI\_ISP\_GetRadialShadingAttr

**【描述】**

获取 Radial LSC 参数

**【语法】**

```
CVI_S32 CVI_ISP_GetRadialShadingAttr(VI_PIPE ViPipe, ISP_RADIAL_SHADING_ATTR_S_
↪ *pstRadialShadingAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRadialShadingAttr	Radial LSC 参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

cv180x & cv181x 不支持此功能

**【举例】**

无

**【相关主题】**

· CVI\_ISP\_SetRadialShadingAttr

### 10.2.3 CVI\_ISP\_SetRadialShadingGainLutAttr

**【描述】**

设置 LSC Radius 形式补偿增益表

**【语法】**

```
CVI_S32 CVI_ISP_SetRadialShadingGainLutAttr(VI_PIPE ViPipe, const ISP_RADIAL_
→SHADING_GAIN_LUT_ATTR_S *pstRadialShadingGainLutAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRadialShading-GainLutAttr	LSC Radius 形式补偿增益表	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

cv180x & cv181x 不支持此功能

**【举例】**

无

**【相关主题】**

· CVI\_ISP\_GetRadialShadingGainLutAttr

## 10.2.4 CVI\_ISP\_GetRadialShadingGainLutAttr

### 【描述】

获取 LSC Radius 形式补偿增益表

### 【语法】

```
CVI_S32 CVI_ISP_GetRadialShadingGainLutAttr(VI_PIPE ViPipe, const ISP_RADIAL_
→SHADING_GAIN_LUT_ATTR_S *pstRadialShadingGainLutAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRadialShading-GainLutAttr	LSC Radius 形式补偿增益表	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

cv180x & cv181x 不支持此功能

### 【举例】

无

### 【相关主题】

- CVI\_ISP\_SetRadialShadingGainLutAttr

## 10.3 数据类型

- ISP\_RADIAL\_SHADING\_MANUAL\_ATTR\_S : Radial LSC 手动参数
- ISP\_RADIAL\_SHADING\_AUTO\_ATTR\_S : Radial LSC 自动参数
- ISP\_RADIAL\_SHADING\_ATTR\_S : Radial LSC 参数
- ISP\_RADIAL\_SHADING\_GAIN\_LUT\_S : Radial LSC Radius 形式补偿增益表细项
- ISP\_RADIAL\_SHADING\_GAIN\_LUT\_ATTR\_S : LSC Radius 形式补偿增益表



### 10.3.1 ISP\_RADIAL\_SHADING\_MANUAL\_ATTR\_S

**【说明】**

Radial LSC 手动参数

**【定义】**

```
typedef struct _ISP_RADIAL_SHADING_MANUAL_ATTR_S {  
    CVI_U16 RadiusStr;  
    CVI_U16 RadiusIRStr;  
} ISP_RADIAL_SHADING_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
RadiusStr	LSC 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
RadiusIRStr	LSC IR 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

- CVI\_ISP\_SetRadialShadingAttr
- CVI\_ISP\_GetRadialShadingAttr

### 10.3.2 ISP\_RADIAL\_SHADING\_AUTO\_ATTR\_S

**【说明】**

Radial LSC 自动参数

**【定义】**

```
typedef struct _ISP_RADIAL_SHADING_AUTO_ATTR_S {  
    CVI_U16 RadiusStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 RadiusIRStr[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_RADIAL_SHADING_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
RadiusStr	LSC 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
RadiusIRStr	LSC IR 补偿强度 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

- CVI\_ISP\_SetRadialShadingAttr
- CVI\_ISP\_GetRadialShadingAttr

### 10.3.3 ISP\_RADIAL\_SHADING\_ATTR\_S

**【说明】**

Radial LSC 参数

**【定义】**

```
typedef struct _ISP_RADIAL_SHADING_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_U16 CenterX;  
    CVI_U16 CenterY;  
    CVI_U16 RadiusScaleRGB;  
    CVI_U16 RadiusScaleIR;  
    ISP_RADIAL_SHADING_MANUAL_ATTR_S stManual;  
    ISP_RADIAL_SHADING_AUTO_ATTR_S stAuto;  
} ISP_RADIAL_SHADING_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	LSC 功能使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	选择手动或自动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
CenterX	图像传感器镜心 X 方向坐标 取值范围: [0x0, 0x1fff] 数据类型: CVI_U16
CenterY	图像传感器镜心 Y 方向坐标 取值范围: [0x0, 0x1fff] 数据类型: CVI_U16
RadiusScaleRGB	RGB Radius 与补偿增益表标准化系数 取值范围: [0x0, 0x7fff] 数据类型: CVI_U16
RadiusScaleIR	IR Radius 与补偿增益表标准化系数 取值范围: [0x0, 0x7fff] 数据类型: CVI_U16
stManual	手动参数
stAuto	自动参数

**【注意事项】**

无

**【相关数据类型及接口】**

- CVI\_ISP\_SetRadialShadingAttr
- CVI\_ISP\_GetRadialShadingAttr

### 10.3.4 ISP\_RADIAL\_SHADING\_GAIN\_LUT\_S

**【说明】**

LSC Radius 形式补偿增益表细项

**【定义】**

```
typedef struct _ISP_RADIAL_SHADING_GAIN_LUT_S {  
    CVI_U16 ColorTemperature;  
    CVI_U16 RGain[ISP_RLSC_WINDOW_SIZE];  
    CVI_U16 GGain[ISP_RLSC_WINDOW_SIZE];  
    CVI_U16 BGain[ISP_RLSC_WINDOW_SIZE];  
    CVI_U16 IrGain[ISP_RLSC_WINDOW_SIZE];  
} ISP_RADIAL_SHADING_GAIN_LUT_S;
```

## 【成员】

成员名称	描述
ColorTemperature	色温自适应 LSC 补偿增益表所对应之色温，单位为 K 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
RGain	LSC Radius 形式红色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
GGain	LSC Radius 形式绿色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
BGain	LSC Radius 形式蓝色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
IrGain	LSC Radius 形式 IR 色通道增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

## 【注意事项】

无

## 【相关数据类型及接口】

- CVI\_ISP\_SetRadialShadingGainLutAttr
- CVI\_ISP\_GetRadialShadingGainLutAttr

### 10.3.5 ISP\_RADIAL\_SHADING\_GAIN\_LUT\_ATTR\_S

## 【说明】

LSC Radius 形式补偿增益表

## 【定义】

```
typedef struct _ISP_RADIAL_SHADING_GAIN_LUT_ATTR_S {  
    CVI_U8 Size;  
    CVI_U16 ColorTemperature[ISP_RLSC_COLOR_TEMPERATURE_SIZE];  
    ISP_RADIAL_SHADING_GAIN_LUT_S RLscGainLut[ISP_RLSC_COLOR_TEMPERATURE_SIZE];  
} ISP_RADIAL_SHADING_GAIN_LUT_ATTR_S;
```

## 【成员】

成员名称	描述
Size	色温自适应 LSC 补偿增益表数量 取值范围: [0x1, 0x7] 数据类型: CVI_U8
ColorTemperature	色温自适应 LSC 补偿增益表所对应之色温, 单位为 K 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
RLscGainLut	LSC Radius 形式补偿增益表细项 取值范围: [0x0, 0xffff] 数据类型: CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

- CVI\_ISP\_SetRadialShadingGainLutAttr
- CVI\_ISP\_GetRadialShadingGainLutAttr

# 11 CCM

## 11.1 功能描述

CCM 为 Color Correction Matrix 的简称。因为 Sensor 在 RGB 的响应与人眼不同，所以需要使  
用此矩阵做转换，让前端捕获的图片亮度与人眼一致。

## 11.2 API 参考

- CVI\_ISP\_SetCCMAAttr: 设置色彩校正矩阵属性参数
- CVI\_ISP\_GetCCMAAttr: 获取色彩校正矩阵属性参数
- CVI\_ISP\_SetCCMSaturationAttr: 设置色彩饱和度属性参数
- CVI\_ISP\_GetCCMSaturationAttr: 获取色彩饱和度属性参数

### 11.2.1 CVI\_ISP\_SetCCMAAttr

#### 【描述】

设置色彩校正矩阵属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCCMAAttr(VI_PIPE ViPipe, const ISP_CCM_ATTR_S *pstCCMAAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCCMAAttr	色彩校正矩阵属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetCCMAAttr](#)

## 11.2.2 CVI\_ISP\_GetCCMAAttr

**【描述】**

获取色彩校正矩阵属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCCMAAttr(VI_PIPE ViPipe, ISP_CCM_ATTR_S *pstCCMAAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCCMAAttr	色彩校正矩阵属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetCCMAAttr](#)

### 11.2.3 CVI\_ISP\_SetCCMSaturationAttr

**【描述】**

设置色彩饱和度属性参数

**【语法】**

```
CVI_S32 CVI_ISP_SetCCMSaturationAttr (VI_PIPE ViPipe, const ISP_CCM_SATURATION_
→ATTR_S * pstCCMSaturationAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCCMSaturationAttr	色彩饱和度属性参数	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_GetCCMSaturationAttr

### 11.2.4 CVI\_ISP\_GetCCMSaturationAttr

**【描述】**

获取色彩饱和度属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCCMSaturationAttr (VI_PIPE ViPipe, const ISP_CCM_SATURATION_
→ATTR_S *pstCCMSaturationAttr);
```



**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCCMSaturationAttr	色彩饱和度属性参数	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvl\_isp.h, cvl\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetCCMSaturationAttr](#)

## 11.3 数据类型

- [ISP\\_COLORMATRIX\\_ATTR\\_S](#): 色彩校正矩阵
- [ISP\\_CCM\\_MANUAL\\_ATTR\\_S](#): 色彩校正矩阵属性手动参数
- [ISP\\_CCM\\_AUTO\\_ATTR\\_S](#): 色彩校正矩阵属性自动参数
- [ISP\\_CCM\\_ATTR\\_S](#): 色彩校正矩阵属性参数
- [ISP\\_CCM\\_SATURATION\\_MANUAL\\_ATTR\\_S](#): 色彩饱和度属性手动参数
- [ISP\\_CCM\\_SATURATION\\_AUTO\\_ATTR\\_S](#): 色彩饱和度属性自动参数
- [ISP\\_CCM\\_SATURATION\\_ATTR\\_S](#): 色彩饱和度属性参数

### 11.3.1 ISP\_COLORMATRIX\_ATTR\_S

**【说明】**

色彩校正矩阵

**【定义】**

```
typedef struct _ISP_COLORMATRIX_ATTR_S {  
    CVI_U16 ColorTemp;  
    CVI_S16 CCM[9];  
} ISP_COLORMATRIX_ATTR_S;
```

**【成员】**

成员名称	描述
ColorTemp	此色彩校正矩阵适用的色温 取值范围：[0x1f4, 0x7530] 数据类型：CVI_U16
CCM[9]	3x3 色彩校正矩阵实际内容 取值范围：[0x-2000, 0x1fff] 数据类型：CVI_S16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCCMAAttr

CVI\_ISP\_GetCCMAAttr

### 11.3.2 ISP\_CCM\_MANUAL\_ATTR\_S

**【说明】**

色彩校正矩阵属性手动参数

**【定义】**

```
typedef struct _ISP_CCM_MANUAL_ATTR_S {  
    CVI_U8 SatEnable;  
    CVI_S16 CCM[9];  
} ISP_CCM_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
SatEnable	手动模式下，饱和度是否生效。 取值范围：[0, 1] 数据类型：CVI_BOOL
CCM[9]	取值范围：[0x-2000, 0x1fff] 数据类型：CVI_S16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCCMAAttr

CVI\_ISP\_GetCCMAAttr

### 11.3.3 ISP\_CCM\_AUTO\_ATTR\_S

**【说明】**

色彩校正矩阵属性自动参数

**【定义】**

```
typedef struct _ISP_CCM_AUTO_ATTR_S {  
    CVI_U8 ISOActEnable;  
    CVI_U8 TempActEnable;  
    CVI_U8 CCMTabNum;  
    ISP_COLORMATRIX_ATTR_S CCMTab[7];  
} ISP_CCM_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
ISOActEnable	低照度下 CCM Bypass 功能使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
TempActEnable	高低色温下 CCM Bypass 功能使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
CCMTabNum	当前配置的 CCM 矩阵个数。 取值范围：[0x3, 0x7] 数据类型：CVI_U8
CCMTab	不同色温下的颜色校正矩阵

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCCMAAttr

CVI\_ISP\_GetCCMAAttr

## 11.3.4 ISP\_CCM\_ATTR\_S

**【说明】**

色彩校正矩阵属性参数

**【定义】**

```
typedef struct _ISP_CCM_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_CCM_MANUAL_ATTR_S stManual;  
    ISP_CCM_AUTO_ATTR_S stAuto;  
} ISP_CCM_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	CCM 模块使能。 0: 关闭。 1: 使能。取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	选择手动或自动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stAuto	自动参数
stManual	手动参数

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCCMAAttr

CVI\_ISP\_GetCCMAAttr

### 11.3.5 ISP\_CCM\_SATURATION\_MANUAL\_ATTR\_S

**【说明】**

色彩饱和度属性手动参数

**【定义】**

```
typedef struct _ISP_CCM_SATURATION_MANUAL_ATTR_S {  
    CVI_U8 SaturationLE;  
    CVI_U8 SaturationSE;  
} ISP_CCM_SATURATION_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
SaturationLE	长曝饱和度 取值范围：[0, 0xff] 数据类型：CVI_U8
SaturationSE	短曝饱和度 取值范围：[0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCCMSaturationAttr

CVI\_ISP\_GetCCMSaturationAttr

### 11.3.6 ISP\_CCM\_SATURATION\_AUTO\_ATTR\_S

**【说明】**

色彩饱和度属性自动参数

**【定义】**

```
typedef struct _ISP_CCM_SATURATION_AUTO_ATTR_S {  
    CVI_U8 SaturationLE[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 SaturationSE[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CCM_SATURATION_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
SaturationLE	长曝饱和度 取值范围: [0, 0xff] 数据类型: CVI_U8
SaturationSE	短曝饱和度 取值范围: [0, 0xff] 数据类型: CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

[CVI\\_ISP\\_SetCCMSaturationAttr](#)

[CVI\\_ISP\\_GetCCMSaturationAttr](#)

### 11.3.7 ISP\_CCM\_SATURATION\_ATTR\_S

**【说明】**

色彩饱和度属性参数

**【定义】**

```
typedef struct _ISP_CCM_SATURATION_ATTR_SS {  
    ISP_CCM_SATURATION_MANUAL_ATTR_S stManual;  
    ISP_CCM_SATURATION_AUTO_ATTR_S stAuto;  
} ISP_CCM_SATURATION_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动参数
stAuto	自动参数

**【注意事项】**

无

**【相关数据类型及接口】**

[CVI\\_ISP\\_SetCCMSaturationAttr](#)

[CVI\\_ISP\\_GetCCMSaturationAttr](#)

# 12 Noise profile

## 12.1 功能描述

记录各种 Sensor Gain 的噪声情形。一般由校正程序获得，不需特别调整。

## 12.2 API 参考

- `CVI_ISP_SetNoiseProfileAttr`: 设置 Noise Profile 属性参数
- `CVI_ISP_GetNoiseProfileAttr`: 获取 Noise Profile 属性参数

### 12.2.1 CVI\_ISP\_SetNoiseProfileAttr

#### 【描述】

设置 Noise Profile 属性参数

#### 【语法】

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNoiseProfileAttr	Noise Profile 属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetNoiseProfileAttr](#)

## 12.2.2 CVI\_ISP\_GetNoiseProfileAttr

**【描述】**

获取 Noise Profile 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetNoiseProfileAttr(VI_PIPE ViPipe, const ISP_CMOS_NOISE_
↪ CALIBRATION_S *pstNoiseProfileAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNoiseProfileAttr	Noise Profile 属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetNoiseProfileAttr](#)



## 12.3 数据类型

- `ISP_CMOS_NOISE_CALIBRATION_S`: Noise Profile 属性参数

### 12.3.1 `ISP_CMOS_NOISE_CALIBRATION_S`

#### 【说明】

Noise Profile 属性参数

#### 【定义】

```
typedef struct cviISP_CMOS_NOISE_CALIBRATION_S {  
    CVI_FLOAT CalibrationCoef[NOISE_PROFILE_ISO_NUM][NOISE_PROFILE_CHANNEL_  
→NUM][NOISE_PROFILE_LEVEL_NUM];  
} ISP_CMOS_NOISE_CALIBRATION_S;
```

#### 【成员】

成员名称	描述
CalibrationCoef [16][4][2]	Raw 数据噪声模型，分 R/Gr/Gb/B 信道，浮点数据形式 取值范围：[-2147483648,2147483647] 数据类型：CVI_FLOAT

#### 【注意事项】

无

#### 【相关数据类型及接口】

`CVI_ISP_SetNoiseProfileAttr`

`CVI_ISP_GetNoiseProfileAttr`

# 13 BNR

## 13.1 功能描述

在 Bayer Domain 进行的空域去噪算法。

## 13.2 API 参考

- `CVI_ISP_SetNRAttr`: 设置 Bayer 降噪参数属性
- `CVI_ISP_GetNRAttr`: 获取 Bayer 降噪参数属性
- `CVI_ISP_SetNRFilterAttr`: 设置 Bayer 降噪滤波器属性
- `CVI_ISP_GetNRFilterAttr`: 获取 Bayer 降噪滤波器属性

### 13.2.1 CVI\_ISP\_SetNRAttr

#### 【描述】

设置 Bayer 降噪参数属性

#### 【语法】

```
CVI_S32 CVI_ISP_SetNRAttr(VI_PIPE ViPipe, const ISP_NR_ATTR_S *pstNRAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNRAttr	Bayer 降噪参数属性	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetNRAttr](#)

## 13.2.2 CVI\_ISP\_GetNRAttr

**【描述】**

获取 NR 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetNRAttr(VI_PIPE ViPipe, ISP_NR_ATTR_S *pstNRAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNRAttr	Bayer 降噪参数属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetNRAttr](#)

### 13.2.3 CVI\_ISP\_SetNRFilterAttr

**【描述】**

设置 Bayer 降噪滤波器属性

**【语法】**

```
CVI_S32 CVI_ISP_SetNRFilterAttr(VI_PIPE ViPipe, const ISP_NR_FILTER_ATTR_S_
↪*pstNRFilterAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNRFilterAttr	Bayer 降噪滤波器属性	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_GetNRFilterAttr

### 13.2.4 CVI\_ISP\_GetNRFilterAttr

**【描述】**

获取 Bayer 降噪滤波器属性

**【语法】**

```
CVI_S32 CVI_ISP_GetNRFilterAttr(VI_PIPE ViPipe, ISP_NR_FILTER_ATTR_S_
↪*pstNRFilterAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstNRFilterAttr	Bayer 降噪滤波器属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

`CVI_ISP_SetNRFilterAttr`

## 13.3 数据类型

- `ISP_NR_MANUAL_ATTR_S`: Bayer 降噪参数手动属性
- `ISP_NR_AUTO_ATTR_S`: Bayer 降噪参数自动属性
- `ISP_NR_ATTR_S`: Bayer 降噪参数属性
- `ISP_NR_FILTER_MANUAL_ATTR_S`: Bayer 降噪滤波器手动属性
- `ISP_NR_FILTER_AUTO_ATTR_S`: Bayer 降噪滤波器自动属性
- `ISP_NR_FILTER_ATTR_S`: Bayer 降噪滤波器属性
- `ISP_RLSC_MANUAL_ATTR_S`: Bayer 降噪 RLSC 手动属性
- `ISP_RLSC_AUTO_ATTR_S`: Bayer 降噪 RLSC 自动属性
- `ISP_RLSC_ATTR_S`: Bayer 降噪 RLSC 属性

### 13.3.1 ISP\_NR\_MANUAL\_ATTR\_S

**【说明】**

Bayer 降噪参数手动属性

**【定义】**

```
typedef struct _ISP_NR_MANUAL_ATTR_S {  
    CVI_U8 WindowType;  
    CVI_U8 DetailSmoothMode;  
    CVI_U8 NoiseSuppressStr;  
    CVI_U8 FilterType;  
    CVI_U8 NoiseSuppressStrMode;  
} ISP_NR_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
WindowType	去噪滤波局域程度。其值越小，作用越局域。 取值范围：[0x0, 0xb] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0: 关闭。 1: 使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8
NoiseSuppressStr	噪声抑制强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterType	去噪滤波器强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressStrMode	亮噪去噪强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetNRAttr

CVI\_ISP\_GetNRAttr

### 13.3.2 ISP\_NR\_AUTO\_ATTR\_S

**【说明】**

Bayer 降噪参数自动属性

**【定义】**

```
typedef struct _ISP_NR_AUTO_ATTR_S {  
    CVI_U8 WindowType[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DetailSmoothMode[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 NoiseSuppressStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 FilterType[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 NoiseSuppressStrMode[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_NR_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
WindowType	去噪滤波局域程度。其值越小，作用越局域。 取值范围：[0x0, 0xb] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0: 关闭。 1: 使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8
NoiseSuppressStr	噪声抑制强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterType	去噪滤波器强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressStrMode	亮噪去噪强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetNRAttr

CVI\_ISP\_GetNRAttr

### 13.3.3 ISP\_NR\_ATTR\_S

#### 【说明】

Bayer 降噪参数属性

#### 【定义】

```
typedef struct _ISP_NR_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_BOOL CoringParamEnable;  
    ISP_NR_MANUAL_ATTR_S stManual;  
    ISP_NR_AUTO_ATTR_S stAuto;  
} ISP_NR_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	BNR 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作模式 0: 自动 1: 手动
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好 取值范围: [0x0, 0xff] 数据类型: CVI_U8
CoringParamEnable	Coring 参数使能 取值范围: [0, 1] 数据类型: CVI_BOOL
stManual	Bayer 降噪参数手动属性
stAuto	Bayer 降噪参数自动属性

#### 【注意事项】

无

#### 【相关数据类型及接口】

CVI\_ISP\_SetNRAttr

CVI\_ISP\_GetNRAttr



### 13.3.4 ISP\_NR\_FILTER\_MANUAL\_ATTR\_S

**【说明】**

Bayer 降噪滤波器手动属性

**【定义】**

```
typedef struct _ISP_NR_FILTER_MANUAL_ATTR_S {  
    CVI_U8 LumaStr[8];  
    CVI_U8 VarThr;  
    CVI_U16 CoringWgtLF;  
    CVI_U16 CoringWgtHF;  
    CVI_U8 NonDirFiltStr;  
    CVI_U8 VhDirFiltStr;  
    CVI_U8 AaDirFiltStr;  
    CVI_U16 NpSlopeR; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpSlopeGr; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpSlopeGb; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpSlopeB; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLumaThrR; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLumaThrGr; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLumaThrGb; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLumaThrB; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLowOffsetR; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLowOffsetGr; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLowOffsetGb; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpLowOffsetB; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpHighOffsetR; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpHighOffsetGr; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpHighOffsetGb; /*RW; Range:[0x0, 0x3ff]*/  
    CVI_U16 NpHighOffsetB; /*RW; Range:[0x0, 0x3ff]*/  
} ISP_NR_FILTER_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
LumaStr[8]	参考图像亮度从暗到亮分别调节去噪强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VarThr	侦测边缘的阈值。值越大，判断为边缘的数量越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
CoringWgtLF	调节在低频区域的随机噪声强度。值越大，在低频区域保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
CoringWgtHF	调节在高频区域的随机噪声强度。值越大，在高频区域的保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
NonDirFiltStr	调节在低频区的去噪强度。值越大，在低频区域去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大，在水平和垂直边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
AaDirFiltStr	调节在对角线边缘的去噪强度。值越大，在对角线边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
NpSlopeR	Noise profile 在 R 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpSlopeGr	Noise profile 在 Gr 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8 -
NpSlopeGb	Noise profile 在 Gb 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpSlopeB	Noise profile 在 B 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrR	Noise profile 在 R 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrGr	Noise profile 在 Gr 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrGb	Noise profile 在 Gb 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrB	Noise profile 在 B 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLowOffsetR	Noise profile 在 R 通道可允许的最小 noise level 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetNRFilterAttr

CVI\_ISP\_GetNRFilterAttr

**13.3.5 ISP\_NR\_FILTER\_AUTO\_ATTR\_S****【说明】**

Bayer 降噪滤波器自动属性

**【定义】**

```
typedef struct _ISP_NR_FILTER_AUTO_ATTR_S {
    CVI_U8 LumaStr[8][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 VarThr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 CoringWgtLF[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 CoringWgtHF[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 NonDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 VhDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 AaDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpSlopeR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpSlopeGr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpSlopeGb[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpSlopeB[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLumaThrR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLumaThrGr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLumaThrGb[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLumaThrB[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLowOffsetR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLowOffsetGr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLowOffsetGb[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpLowOffsetB[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpHighOffsetR[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpHighOffsetGr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpHighOffsetGb[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 NpHighOffsetB[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_NR_FILTER_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
LumaStr[8]	参考图像亮度从暗到亮分别调节去噪强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VarThr	侦测边缘的阈值。值越大，判断为边缘的数量越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
CoringWgtLF	调节在低频区域的随机噪声强度。值越大，在低频区域保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
CoringWgtHF	调节在高频区域的随机噪声强度。值越大，在高频区域的保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
NonDirFiltStr	调节在低频区的去噪强度。值越大，在低频区域去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大，在水平和垂直边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
AaDirFiltStr	调节在对角线边缘的去噪强度。值越大，在对角线边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
NpSlopeR	Noise profile 在 R 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpSlopeGr	Noise profile 在 Gr 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpSlopeGb	Noise profile 在 Gb 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpSlopeB	Noise profile 在 B 通道的斜率 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrR	Noise profile 在 R 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrGr	Noise profile 在 Gr 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrGb	Noise profile 在 Gb 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLumaThrB	Noise profile 在 B 通道的亮度阈值 取值范围：[0x0, 0x3ff] 数据类型：CVI_U8
NpLowOffsetR	Noise profile 在 R 通道可允许的最小 noise

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetNRFilterAttr

CVI\_ISP\_GetNRFilterAttr

### 13.3.6 ISP\_NR\_FILTER\_ATTR\_S

**【说明】**

Bayer 降噪滤波器属性

**【定义】**

```
typedef struct _ISP_NR_FILTER_ATTR_S {  
    CVI_U8 TuningMode;  
    ISP_NR_FILTER_MANUAL_ATTR_S stManual;  
    ISP_NR_FILTER_AUTO_ATTR_S stAuto;  
} ISP_NR_FILTER_ATTR_S;
```

**【成员】**

成员名称	描述
TuningMode	输出调试方案，辅助调节参数。 8: BNR 图像结果。 11: 平坦/边缘侦测图像结果。 12: 垂直边缘侦测图像结果。 13: 水平边缘侦测图像结果。 14: 反对角线边缘侦测图像结果。 15: 对角线边缘侦测图像结果。 取值范围：[0x0, 0xf] 数据类型：CVI_U8
stManual	自动模式属性。
stAuto	手动模式属性。

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetNRFilterAttr

CVI\_ISP\_GetNRFilterAttr

### 13.3.7 ISP\_RLSC\_MANUAL\_ATTR\_S

**【说明】**

Bayer 降噪 RLSC 手动属性

**【定义】**

```
typedef struct cviISP_RLSC_MANUAL_ATTR_S {  
    CVI_U16 RadialStr;  
} ISP_RLSC_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
RadialStr	用来调整 RLSC 标定后的强度。值越大，越接近标定后的强度。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetRLSCAttr

CVI\_ISP\_GetRLSCAttr

### 13.3.8 ISP\_RLSC\_AUTO\_ATTR\_S

**【说明】**

Bayer 降噪 RLSC 自动属性

**【定义】**

```
typedef struct cviISP_RLSC_AUTO_ATTR_S {  
    CVI_U16 RadialStr[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_RLSC_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
RadialStr	用来调整 RLSC 标定后的强度。值越大，越接近标定后的强度。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

**【注意事项】**

无

## 【相关数据类型及接口】

CVI\_ISP\_SetRLSCAttr

CVI\_ISP\_GetRLSCAttr

### 13.3.9 ISP\_RLSC\_ATTR\_S

## 【说明】

Bayer 降噪 RLSC 属性

## 【定义】

```
typedef struct _ISP_RLSC_ATTR_S {  
    CVI_BOOL RlscEnable;  
    CVI_U16 RlscCenterX;  
    CVI_U16 RlscCenterY;  
    ISP_RLSC_MANUAL_ATTR_S stManual;  
    ISP_RLSC_AUTO_ATTR_S stAuto;  
} ISP_RLSC_ATTR_S;
```

## 【成员】

成员名称	描述
RlscEnable	用来调整 RLSC 标定后的强度。值越大，越接近标定后的强度。 取值范围：[0, 1] 数据类型：CVI_BOOL
RlscCenterX	X 中心位置 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
RlscCenterY	Y 中心位置 取值范围：[0x0, 0x7ff] 数据类型：CVI_U16
stManual	Bayer 降噪 RLSC 手动模式属性
stAuto	Bayer 降噪 RLSC 自动模式属性

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetRLSCAttr

CVI\_ISP\_GetRLSCAttr

# 14 YNR

## 14.1 功能描述

在 YUV Domain 针对亮度噪声进行的空域去噪算法。

## 14.2 API 参考

- `CVI_ISP_SetYNRAAttr`: 设置亮度降噪属性参数
- `CVI_ISP_GetYNRAAttr`: 获取亮度降噪属性参数
- `CVI_ISP_SetYNRMotionNRAAttr`: 设置亮度降噪移动物体属性参数
- `CVI_ISP_GetYNRMotionNRAAttr`: 获取亮度降噪移动物体属性参数
- `CVI_ISP_SetYNRFilterAttr`: 设置亮度降噪滤波器属性参数
- `CVI_ISP_GetYNRFilterAttr`: 获取亮度降噪滤波器属性参数

### 14.2.1 CVI\_ISP\_SetYNRAAttr

#### 【描述】

设置亮度降噪属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetYNRAAttr(VI_PIPE ViPipe, const ISP_YNR_ATTR_S *pstYNRAAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRAAttr	亮度降噪属性参数	输入

#### 【返回值】



返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetYNRAAttr](#)

## 14.2.2 CVI\_ISP\_GetYNRAAttr

**【描述】**

获取亮度降噪属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetYNRAAttr(VI_PIPE ViPipe, ISP_YNR_ATTR_S *pstYNRAAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRAAttr	亮度降噪属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

【相关主题】

CVI\_ISP\_SetYNRAAttr

14.2.3 CVI\_ISP\_SetYNRMotionNRAttr

【描述】

设置亮度降噪移动物体属性参数

【语法】

```
CVI_S32 CVI_ISP_SetYNRMotionNRAttr(VI_PIPE ViPipe, const ISP_YNR_MOTION_NR_
↪ATTR_S *pstYNRMotionNRAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRMotion-NRAAttr	设置亮度降噪移动物体属性参数	输入

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

【注意】

无

【举例】

无

【相关主题】

CVI\_ISP\_GetYNRMotionNRAttr

## 14.2.4 CVI\_ISP\_GetYNRMotionNRAttr

### 【描述】

获取亮度降噪移动物体属性参数

### 【语法】

```
CVI_S32 CVI_ISP_GetYNRMotionNRAttr(VI_PIPE ViPipe, ISP_YNR_MOTION_NR_ATTR_S_
↪ *pstYNRMotionNRAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRMotionNRAttr	设置亮度降噪移动物体属性参数	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无

### 【举例】

无

### 【相关主题】

[CVI\\_ISP\\_SetYNRMotionNRAttr](#)

## 14.2.5 CVI\_ISP\_SetYNRFilterAttr

### 【描述】

设置亮度降噪滤波器属性参数

### 【语法】

```
CVI_S32 CVI_ISP_SetYNRFilterAttr(VI_PIPE ViPipe, const ISP_YNR_FILTER_ATTR_S_
↪ *pstYNRFilterAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRFilterAttr	亮度降噪滤波器属性参数	输入

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

## 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

## 【注意】

无

## 【举例】

无

## 【相关主题】

[CVI\\_ISP\\_GetYNRFilterAttr](#)

## 14.2.6 CVI\_ISP\_GetYNRFilterAttr

## 【描述】

获取亮度降噪滤波器属性参数

## 【语法】

```
CVI_S32 CVI_ISP_GetYNRFilterAttr(VI_PIPE ViPipe, ISP_YNR_FILTER_ATTR_S_
→*pstYNRFilterAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYNRFilterAttr	亮度降噪滤波器属性参数	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

`CVI_ISP_SetYNRFilterAttr`

## 14.3 数据类型

- `ISP_YNR_MANUAL_ATTR_S`: 亮度降噪属性手动参数
- `ISP_YNR_AUTO_ATTR_S`: 亮度降噪属性自动参数
- `ISP_YNR_ATTR_S`: 亮度降噪属性参数
- `ISP_YNR_MOTION_NR_MANUAL_ATTR_S`: 亮度降噪移动物体属性手动参数
- `ISP_YNR_MOTION_NR_AUTO_ATTR_S`: 亮度降噪移动物体属性自动参数
- `ISP_YNR_MOTION_NR_ATTR_S`: 亮度降噪移动物体属性参数
- `ISP_YNR_FILTER_MANUAL_ATTR_S`: 亮度降噪滤波器属性手动参数
- `ISP_YNR_FILTER_AUTO_ATTR_S`: 亮度降噪滤波器属性自动参数
- `ISP_YNR_FILTER_ATTR_S`: 亮度降噪滤波器属性参数

### 14.3.1 `ISP_YNR_MANUAL_ATTR_S`

**【说明】**

亮度降噪属性手动参数

**【定义】**

```
typedef struct _ISP_YNR_MANUAL_ATTR_S {  
    CVI_U8 WindowType;  
    CVI_U8 DetailSmoothMode;  
    CVI_U8 NoiseSuppressStr;  
    CVI_U8 FilterType;  
    CVI_U8 NoiseCoringMax;  
    CVI_U8 NoiseCoringBase;  
    CVI_U8 NoiseCoringAdv;  
} ISP_YNR_MANUAL_ATTR_S;
```

## 【成员】

成员名称	描述
WindowType	去噪滤波局域程度。其值越小，作用越局域。 取值范围：[0x0, 0xb] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0: 关闭。 1: 使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8
NoiseSuppressStr	噪声抑制强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterType	去噪滤波器强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringMax	噪声抑制强度可允许之最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringBase	运动区亮度噪声容忍值，运动区的判断与 TNR 运动区侦测联动。值越大，对运动区去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringAdv	静止区亮度噪声容忍值，静止区的判断与 TNR 运动区侦测联动。值越大，对静止区去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetYNRAAttr

CVI\_ISP\_GetYNRAAttr

## 14.3.2 ISP\_YNR\_AUTO\_ATTR\_S

## 【说明】

亮度降噪属性自动参数

## 【定义】

```
typedef struct _ISP_YNR_AUTO_ATTR_S {
    CVI_U8 WindowType[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DetailSmoothMode[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 NoiseSuppressStr[ISP_AUTO_ISO_STRENGTH_NUM];
}
```

(下页继续)

(续上页)

```

CVI_U8 FilterType[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 NoiseCoringMax[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 NoiseCoringBase[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 NoiseCoringAdv[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_YNR_AUTO_ATTR_S;

```

**【成员】**

成员名称	描述
WindowType	去噪滤波局域程度。其值越小，作用越局域。 取值范围：[0x0, 0xb] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0: 关闭。 1: 使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8
NoiseSuppressStr	噪声抑制强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterType	去噪滤波器强度。值越大，亮噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringMax	噪声抑制强度可允许之最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringBase	运动区亮度噪声容忍值，运动区的判断与 TNR 运动区侦测联动。值越大，对运动区去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseCoringAdv	静止区亮度噪声容忍值，静止区的判断与 TNR 运动区侦测联动。值越大，对静止区去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRAAttr

CVI\_ISP\_GetYNRAAttr

### 14.3.3 ISP\_YNR\_ATTR\_S

**【说明】**

亮度降噪属性参数

**【定义】**

```
typedef struct _ISP_YNR_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_BOOL CoringParamEnable;  
    CVI_BOOL FiltModeEnable;  
    CVI_U16 FiltMode;  
    CVI_U8 TuningMode;  
    ISP_YNR_MANUAL_ATTR_S stManual;  
    ISP_YNR_AUTO_ATTR_S stAuto;  
} ISP_YNR_ATTR_S;
```

**【成员】**



成员名称	描述
Enable	YNR 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
CoringParamEnable	控制是否使用手动 coring, 假如为 0 则 NoiseCoringBaseLuma[6] / NoiseCoringBaseOffset[6] / NoiseCoringAdvLuma[6] / NoiseCoringAdvOffset[6] 没有用。 取值范围: [0, 1] 数据类型: CVI_BOOL
FiltModeEnable	滤波器手调混和模式使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
FiltMode	滤波器手调混和权重。 取值范围: [0, 0x100] 数据类型: CVI_U16
TuningMode	输出元文件案, 辅助调节参数。8: YNR 图像结果。11: 平坦/边缘侦测图像结果。(Remove 12, 13, 14, 15) 12: 垂直边缘 侦测图像结果。13: 水平边缘侦测图像结果。14: 反对角线边 缘侦测图像结果。15: 对角线边缘侦测图像结果。 取值范围: [0x0, 0xf] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRAAttr

CVI\_ISP\_GetYNRAAttr

### 14.3.4 ISP\_YNR\_MOTION\_NR\_MANUAL\_ATTR\_S

**【说明】**

亮度降噪移动物体属性手动参数

**【定义】**

```
typedef struct _ISP_YNR_MOTION_NR_MANUAL_ATTR_S {  
    CVI_U8 MotionCoringWgtMax;  
    CVI_U16 MotionYnrLut[16];  
    CVI_U16 MotionCoringWgt[16];  
} ISP_YNR_MOTION_NR_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
MotionCoringWgtMax	针对物体运动区域，可允许保留噪声的最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionYnrLut[16]	调整在不同物体运动量时对应的去亮噪强度，将运动量区分为16阶。值越大，去亮噪强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U16
MotionCoringWgt [16]	调整在不同物体运动量时对应的噪声保留程度，将运动量区分为16阶。值越大，噪声保留越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRMotionNRAttr

CVI\_ISP\_GetYNRMotionNRAttr

### 14.3.5 ISP\_YNR\_MOTION\_NR\_AUTO\_ATTR\_S

**【说明】**

亮度降噪移动物体属性自动参数

**【定义】**

```
typedef struct _ISP_YNR_MOTION_NR_AUTO_ATTR_S {  
    CVI_U8 MotionCoringWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 MotionYnrLut[16][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 MotionCoringWgt[16][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_YNR_MOTION_NR_AUTO_ATTR_S;
```

## 【成员】

成员名称	描述
MotionCoringWgtMax	针对物体运动区域，可允许保留噪声的最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionYnrLut[16]	调整在不同物体运动量时对应的去亮噪强度，将运动量区分为 16 阶。值越大，去亮噪强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U16
MotionCoringWgt [16]	调整在不同物体运动量时对应的噪声保留程度，将运动量区分为 16 阶。值越大，噪声保留越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetYNRMotionNRAttr

CVI\_ISP\_GetYNRMotionNRAttr

### 14.3.6 ISP\_YNR\_MOTION\_NR\_ATTR\_S

## 【说明】

亮度降噪移动物体属性参数

## 【定义】

```
typedef struct ISP_YNR_MOTION_NR_ATTR_S {
    ISP_YNR_MOTION_NR_MANUAL_ATTR_S stManual;
    ISP_YNR_MOTION_NR_AUTO_ATTR_S stAuto;
} ISP_YNR_MOTION_NR_ATTR_S;
```

## 【成员】

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetYNRMotionNRAttr

CVI\_ISP\_GetYNRMotionNRAttr

### 14.3.7 ISP\_YNR\_FILTER\_MANUAL\_ATTR\_S

**【说明】**

亮度降噪滤波器属性手动参数

**【定义】**

```
typedef struct _ISP_YNR_FILTER_MANUAL_ATTR_S {  
    CVI_U8 VarThr;  
    CVI_U16 CoringWgtLF;  
    CVI_U16 CoringWgtHF;  
    CVI_U8 NonDirFiltStr;  
    CVI_U8 VhDirFiltStr;  
    CVI_U8 AaDirFiltStr;  
    CVI_U8 CoringWgtMax;  
    CVI_U16 FilterMode;  
} ISP_YNR_FILTER_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
VarThr	侦测边缘的阈值。值越大，判断为边缘的数量越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
CoringWgtLF	调节在低频区域的随机噪声强度。值越大，在低频区域保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
CoringWgtHF	调节在高频区域的随机噪声强度。值越大，在高频区域的保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
NonDirFiltStr	调节在低频区的去噪强度。值越大，在低频区域去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大，在水平和垂直边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
AaDirFiltStr	调节在对角线边缘的去噪强度。值越大，在对角线边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
CoringWgtMax	可允许保留噪声的最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterMode	滤波器模式。值越小，平坦区保留的噪声均匀性越好。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRFilterAttr

CVI\_ISP\_GetYNRFilterAttr

### 14.3.8 ISP\_YNR\_FILTER\_AUTO\_ATTR\_S

**【说明】**

亮度降噪滤波器属性自动参数

**【定义】**

```
typedef struct ISP_YNR_FILTER_AUTO_ATTR_S {  
    CVI_U8 VarThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 CoringWgtLF[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 CoringWgtHF[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 NonDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 VhDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 AaDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 CoringWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 FilterMode[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_YNR_FILTER_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
VarThr	侦测边缘的阈值。值越大，判断为边缘的数量越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
CoringWgtLF	调节在低频区域的随机噪声强度。值越大，在低频区域保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
CoringWgtHF	调节在高频区域的随机噪声强度。值越大，在高频区域的保留的噪声越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
NonDirFiltStr	调节在低频区的去噪强度。值越大，在低频区域去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大，在水平和垂直边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
AaDirFiltStr	调节在对角线边缘的去噪强度。值越大，在对角线边缘去除的噪声越多。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
CoringWgtMax	可允许保留噪声的最大值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
FilterMode	滤波器模式。值越小，平坦区保留的噪声均匀性越好。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRFilterAttr

CVI\_ISP\_GetYNRFilterAttr

### 14.3.9 ISP\_YNR\_FILTER\_ATTR\_S

**【说明】**

亮度降噪滤波器属性参数

**【定义】**

```
typedef struct _ISP_YNR_FILTER_ATTR_S {  
    ISP_YNR_FILTER_MANUAL_ATTR_S stManual;  
    ISP_YNR_FILTER_AUTO_ATTR_S stAuto;  
} ISP_YNR_FILTER_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetYNRFilterAttr

CVI\_ISP\_GetYNRFilterAttr

# 15 CNR

## 15.1 功能描述

在 YUV Domain 针对色度噪声进行的空域去噪算法。

## 15.2 API 参考

- `CVI_ISP_SetCNRAttr`: 设置色度降噪参数属性
- `CVI_ISP_GetCNRAttr`: 获取色度降噪参数属性
- `CVI_ISP_SetCNRMotionNRAttr`: 设置色度降噪移动物体参数属性
- `CVI_ISP_GetCNRMotionNRAttr`: 获取色度降噪移动物体参数属性

### 15.2.1 CVI\_ISP\_SetCNRAttr

#### 【描述】

设置色度降噪参数属性

#### 【语法】

```
CVI_S32 CVI_ISP_SetCNRAttr(VI_PIPE ViPipe, const ISP_CNR_ATTR_S *pstCNRAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCNRAttr	色度降噪参数属性	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。



**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetCNRAAttr](#)

## 15.2.2 CVI\_ISP\_GetCNRAAttr

**【描述】**

获取色度降噪参数属性

**【语法】**

```
CVI_S32 CVI_ISP_GetCNRAAttr(VI_PIPE ViPipe, ISP_CNR_ATTR_S *pstCNRAAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCNRAAttr	色度降噪参数属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetCNRAAttr](#)

### 15.2.3 CVI\_ISP\_SetCNRMotionNRAttr

**【描述】**

设置色度降噪移动物体参数属性

**【语法】**

```
CVI_S32 CVI_ISP_SetCNRMotionNRAttr(VI_PIPE ViPipe, const ISP_CNR_MOTION_NR_
→ATTR_S *pstCNRMotionNRAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCNRMotion- NRAttr	色度降噪移动物体参数属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_GetCNRMotionNRAttr

### 15.2.4 CVI\_ISP\_GetCNRMotionNRAttr

**【描述】**

获取色度降噪移动物体参数属性

**【语法】**

```
CVI_S32 CVI_ISP_GetCNRMotionNRAttr(VI_PIPE ViPipe, const ISP_CNR_MOTION_NR_
→ATTR_S *pstCNRMotionNRAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCNRMotion-NRAttr	色度降噪移动物体参数属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvl\_isp.h, cvl\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_SetCNRMotionNRAttr

## 15.3 数据类型

- ISP\_CNR\_MANUAL\_ATTR\_S: 色度降噪参数手动属性
- ISP\_CNR\_AUTO\_ATTR\_S: 色度降噪参数自动属性
- ISP\_CNR\_ATTR\_S: 色度降噪参数属性
- ISP\_CNR\_MOTION\_NR\_MANUAL\_ATTR\_S: 色度降噪移动物体参数手动属性
- ISP\_CNR\_MOTION\_NR\_AUTO\_ATTR\_S: 色度降噪移动物体参数自动属性
- ISP\_CNR\_MOTION\_NR\_ATTR\_S: 色度降噪移动物体参数属性

### 15.3.1 ISP\_CNR\_MANUAL\_ATTR\_S

**【说明】**

色度降噪参数手动属性

**【定义】**

```
typedef struct _ISP_CNR_MANUAL_ATTR_S {  
    CVI_U8 CnrStr;  
    CVI_U8 NoiseSuppressStr;  
    CVI_U8 NoiseSuppressGain;  
    CVI_U8 FilterType;  
    CVI_U8 MotionNrStr;  
    CVI_U8 LumaWgt;  
    CVI_U8 DetailSmoothMode;  
} ISP_CNR_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
CnrStr	色噪去噪强度。值越大，色噪去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressStr	色噪抑制强度。值越大，色噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressGain	色噪抑制强度增益。值越小，色噪去噪强度越大。 取值范围：[0x1, 0x8] 数据类型：CVI_U8
FilterType	色噪去噪滤波器强度。值越大，色噪去除强度越大。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
MotionNrStr	调节物体运动区域的色噪去噪强度。值越大，运动区域的色噪越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgt	参考亮度的色噪去噪权重。值越大，亮度影响色噪去噪的强度越大。 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0：关闭。 1：使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCNRAAttr

CVI\_ISP\_GetCNRAAttr

### 15.3.2 ISP\_CNR\_AUTO\_ATTR\_S

#### 【说明】

色度降噪参数自动属性

#### 【定义】

```
typedef struct _ISP_CNR_AUTO_ATTR_S {  
    CVI_U8 CnrStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 NoiseSuppressStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 NoiseSuppressGain[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 FilterType[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 MotionNrStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgt[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DetailSmoothMode[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CNR_AUTO_ATTR_S;
```

#### 【成员】

成员名称	描述
CnrStr	色噪去噪强度。值越大，色噪去噪强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressStr	色噪抑制强度。值越大，色噪去除强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
NoiseSuppressGain	色噪抑制强度增益。值越小，色噪去噪强度越大。 取值范围：[0x1, 0x8] 数据类型：CVI_U8
FilterType	色噪去噪滤波器强度。值越大，色噪去除强度越大。 取值范围：[0x0, 0x1f] 数据类型：CVI_U8
MotionNrStr	调节物体运动区域的色噪去噪强度。值越大，运动区域的色噪越少。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgt	参考亮度的色噪去噪权重。值越大，亮度影响色噪去噪的强度越大。 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DetailSmoothMode	去噪细节平滑功能使能。 0: 关闭。 1: 使能。 取值范围：[0x0, 0x1] 数据类型：CVI_U8

【注意事项】

无

【相关数据类型及接口】

CVI\_ISP\_SetCNRAAttr

CVI\_ISP\_GetCNRAAttr

15.3.3 ISP\_CNR\_ATTR\_S

【说明】

色度降噪参数属性

【定义】

```
typedef struct _ISP_CNR_ATTR_S {
    CVI_BOOL Enable;
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval;
    ISP_CNR_MANUAL_ATTR_S stManual;
    ISP_CNR_AUTO_ATTR_S stAuto;
} ISP_CNR_ATTR_S;
```

【成员】

成员名称	描述
Enable	CNR 模块使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

【注意事项】

无

【相关数据类型及接口】

CVI\_ISP\_SetCNRAAttr

CVI\_ISP\_GetCNRAAttr

### 15.3.4 ISP\_CNR\_MOTION\_NR\_MANUAL\_ATTR\_S

**【说明】**

色度降噪移动物体参数手动属性

**【定义】**

```
typedef struct _ISP_CNR_MOTION_NR_MANUAL_ATTR_S {  
    CVI_U8 MotionCnrCoringLut[16]  
    CVI_U8 MotionCnrStrLut[16];  
} ISP_CNR_MOTION_NR_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
MotionCnrCoringLut[16]	使用 LUT 调整不同物体运动量时对应的色噪抑制强度，将运动量区分为 16 阶。值越大，去色噪强度越强 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCnrStrLut[16]	使用 LUT 调整不同物体运动量时对应的去色噪强度，将运动量区分为 16 阶。值越大，去色噪强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCNRMotionNRAttr

CVI\_ISP\_GetCNRMotionNRAttr

### 15.3.5 ISP\_CNR\_MOTION\_NR\_AUTO\_ATTR\_S

**【说明】**

色度降噪移动物体参数自动属性

**【定义】**

```
typedef struct _ISP_CNR_MOTION_NR_AUTO_ATTR_S {  
    CVI_U8 MotionCnrCoringLut[16][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 MotionCnrStrLut[16][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CNR_MOTION_NR_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
MotionCnrStrLut[16]	使用 LUT 调整不同物体运动量时对应的去色噪强度，将运动量区分为 16 阶。值越大，去色噪强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCNRMotionNRAttr

CVI\_ISP\_GetCNRMotionNRAttr

### 15.3.6 ISP\_CNR\_MOTION\_NR\_ATTR\_S

**【说明】**

色度降噪移动物体参数属性

**【定义】**

```
typedef struct _ISP_CNR_MOTION_NR_ATTR_S {  
    CVI_BOOL MotionCnrEnable;  
    ISP_CNR_MOTION_NR_MANUAL_ATTR_S stManual;  
    ISP_CNR_MOTION_NR_AUTO_ATTR_S stAuto;  
} ISP_CNR_MOTION_NR_ATTR_S;
```

**【成员】**

成员名称	描述
MotionCnrEnable	参考物体运动量来调整去色噪强度功能使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCNRMotionNRAttr

CVI\_ISP\_GetCNRMotionNRAttr



# 16 TNR

## 16.1 功能描述

在 YUV Domain 进行的时域去噪算法，又称为 3DNR。

## 16.2 API 参考

- CVI\_ISP\_SetTNRAttr: 设置 TNR 属性参数
- CVI\_ISP\_GetTNRAttr: 获取 TNR 属性参数
- CVI\_ISP\_SetTNRNoiseModelAttr: 设置 TNR Noise Model 属性参数
- CVI\_ISP\_GetTNRNoiseModelAttr: 获取 TNR Noise Model 属性参数
- CVI\_ISP\_SetTNRlumaMotionAttr: 设置 TNR 亮度对强度增益特性表
- CVI\_ISP\_GetTNRlumaMotionAttr: 获取 TNR 亮度对强度增益特性表
- CVI\_ISP\_SetTNRGhostAttr: 设置 TNR 动量对拖尾消除程度特性表
- CVI\_ISP\_GetTNRGhostAttr: 获取 TNR 动量对拖尾消除程度特性表
- CVI\_ISP\_SetTNRMtPrtAttr: 设置 TNR 动量保护属性参数
- CVI\_ISP\_GetTNRMtPrtAttr: 获取 TNR 动量保护属性参数
- CVI\_ISP\_SetTNRMotionAdaptAttr: 设置 TNR 动量对强度增益特性表
- CVI\_ISP\_GetTNRMotionAdaptAttr: 获取 TNR 动量对强度增益特性表

### 16.2.1 CVI\_ISP\_SetTNRAttr

#### 【描述】

设置 TNR 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetTNRAttr(VI_PIPE ViPipe, const ISP_TNR_ATTR_S *pstTNRAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRAttr	TNR 属性参数	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

```
VI_PIPE ViPipe = 0;
ISP_TNR_ATTR_S stAttr;
CVI_ISP_GetTNRAttr(ViPipe, &stAttr);
stAttr.enOpType = OP_TYPE_AUTO;
CVI_ISP_SetTNRAttr(ViPipe, &stAttr);
```

**【相关主题】**

[CVI\\_ISP\\_GetTNRAttr](#)

## 16.2.2 CVI\_ISP\_GetTNRAttr

**【描述】**

获取 TNR 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetTNRAttr(VI_PIPE ViPipe, ISP_TNR_ATTR_S *pstTNRAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRAttr	TNR 属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_SetTNRAttr

### 16.2.3 CVI\_ISP\_SetTNRNoiseModelAttr

**【描述】**

设置 TNR Noise Model 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_SetTNRNoiseModelAttr(VI_PIPE ViPipe, const ISP_TNR_NOISE_MODEL_
→ATTR_S *pstTNRNoiseModelAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRNoiseModelAttr	TNR Noise Model 属性参数	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

【举例】

无

【相关主题】

CVI\_ISP\_GetTNRNoiseModelAttr

16.2.4 CVI\_ISP\_GetTNRNoiseModelAttr

【描述】

获取 TNR Noise Model 属性参数

【语法】

```
CVI_S32 CVI_ISP_GetTNRNoiseModelAttr(VI_PIPE ViPipe, ISP_TNR_NOISE_MODEL_
→ATTR_S *pstTNRNoiseModelAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRNoiseModelAttr	TNR Noise Model 属性参数	输出

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

【注意】

无

【举例】

无

【相关主题】

CVI\_ISP\_SetTNRNoiseModelAttr

## 16.2.5 CVI\_ISP\_SetTNR\_LumaMotionAttr

### 【描述】

设置 TNR 亮度对强度增益特性表

### 【语法】

```
CVI_S32 CVI_ISP_SetTNR_LumaMotionAttr(VI_PIPE ViPipe, const ISP_TNR_LUMA_MOTION_
→ATTR_S *pstTNR_LumaMotionAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNR_LumaMotionAttr	TNR 亮度对强度增益特性表	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无

### 【举例】

无

### 【相关主题】

[CVI\\_ISP\\_GetTNR\\_LumaMotionAttr](#)

## 16.2.6 CVI\_ISP\_GetTNR\_LumaMotionAttr

### 【描述】

获取 TNR 亮度对强度增益特性表

### 【语法】

```
CVI_S32 CVI_ISP_GetTNR_LumaMotionAttr(VI_PIPE ViPipe, const ISP_TNR_LUMA_MOTION_
→ATTR_S *pstTNR_LumaMotionAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNR_LumaMotionAttr	TNR 亮度对强度增益特性表	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

## 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

## 【注意】

无

## 【举例】

无

## 【相关主题】

`CVI_ISP_SetTNR_LumaMotionAttr`

## 16.2.7 CVI\_ISP\_SetTNRGhostAttr

## 【描述】

设置 TNR 动量对拖尾消除程度特性表

## 【语法】

```
CVI_S32 CVI_ISP_SetTNRGhostAttr(VI_PIPE ViPipe, const ISP_TNR_GHOST_ATTR_S_
→ *pstTNRGhostAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRGhostAttr	TNR 动量对拖尾消除程度特性表	输入

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetTNRGhostAttr](#)

## 16.2.8 CVI\_ISP\_GetTNRGhostAttr

**【描述】**

获取 TNR 动量对拖尾消除程度特性表

**【语法】**

```
CVI_S32 CVI_ISP_GetTNRGhostAttr(VI_PIPE ViPipe, ISP_TNR_GHOST_ATTR_S_
↪*pstTNRGhostAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRGhostAttr	TNR 动量对拖尾消除程度特性表	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_SetTNRGhostAttr

## 16.2.9 CVI\_ISP\_SetTNRMtPrtAttr

**【描述】**

设置 TNR 动量保护属性参数

**【语法】**

```
CVI_S32 CVI_ISP_SetTNRMtPrtAttr(VI_PIPE ViPipe, const ISP_TNR_MT_PRT_ATTR_S_
↪*pstTNRMtPrtAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRMtPrtAttr	TNR 动量保护属性参数	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_GetTNRMtPrtAttr

## 16.2.10 CVI\_ISP\_GetTNRMtPrtAttr

**【描述】**

获取 TNR 动量保护属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetTNRMtPrtAttr(VI_PIPE ViPipe, ISP_TNR_MT_PRT_ATTR_S_
↪*pstTNRMtPrtAttr);
```



## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRMtPrtAttr	TNR 动量保护属性参数	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

## 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

## 【注意】

无

## 【举例】

无

## 【相关主题】

[CVI\\_ISP\\_SetTNRMtPrtAttr](#)

## 16.2.11 CVI\_ISP\_SetTNRMotionAdaptAttr

## 【描述】

设置 TNR 动量对强度增益特性表

## 【语法】

```
CVI_S32 CVI_ISP_SetTNRMotionAdaptAttr(VI_PIPE ViPipe, const ISP_TNR_MOTION_
↪ADAPT_ATTR_S *pstTNRMotionAdaptAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRMotionAdaptAttr	TNR 动量对强度增益特性表	输入

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetTNRMotionAdaptAttr](#)

## 16.2.12 CVI\_ISP\_GetTNRMotionAdaptAttr

**【描述】**

获取 TNR 动量对强度增益特性表

**【语法】**

```
CVI_S32 CVI_ISP_GetTNRMotionAdaptAttr(VI_PIPE ViPipe, ISP_TNR_MOTION_ADAPT_
↪ATTR_S *pstTNRMotionAdaptAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstTNRMotionAdaptAttr	TNR 动量对强度增益特性表	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_SetTNRMotionAdaptAttr

## 16.3 数据类型

- ISP\_TNR\_MANUAL\_ATTR\_S: TNR 属性手动参数
- ISP\_TNR\_AUTO\_ATTR\_S: TNR 属性自动参数
- ISP\_TNR\_ATTR\_S: TNR 属性参数
- ISP\_TNR\_NOISE\_MODEL\_MANUAL\_ATTR\_S: TNR Noise Model 属性手动参数
- ISP\_TNR\_NOISE\_MODEL\_AUTO\_ATTR\_S: TNR Noise Model 属性自动参数
- ISP\_TNR\_NOISE\_MODEL\_ATTR\_S: TNR Noise Model 属性参数
- ISP\_TNR\_LUMA\_MOTION\_MANUAL\_ATTR\_S: 手动模式 TNR 亮度对强度增益特性表
- ISP\_TNR\_LUMA\_MOTION\_AUTO\_ATTR\_S: 自动模式 TNR 亮度对强度增益特性表
- ISP\_TNR\_LUMA\_MOTION\_ATTR\_S: TNR 亮度对强度增益特性表
- ISP\_TNR\_GHOST\_MANUAL\_ATTR\_S: 手动模式动量对拖尾消除程度特性表
- ISP\_TNR\_GHOST\_AUTO\_ATTR\_S: 自动模式动量对拖尾消除程度特性表
- ISP\_TNR\_GHOST\_ATTR\_S: 动量对拖尾消除程度特性表
- ISP\_TNR\_MT\_PRT\_MANUAL\_ATTR\_S: 手动模式 TNR 动量保护属性参数
- ISP\_TNR\_MT\_PRT\_AUTO\_ATTR\_S: 自动模式 TNR 动量保护属性参数
- ISP\_TNR\_MT\_PRT\_ATTR\_S: TNR 动量保护属性参数
- ISP\_TNR\_MOTION\_ADAPT\_MANUAL\_ATTR\_S: 手动模式动量对强度增益特性表
- ISP\_TNR\_MOTION\_ADAPT\_AUTO\_ATTR\_S: 自动模式动量对强度增益特性表
- ISP\_TNR\_MOTION\_ADAPT\_ATTR\_S: 动量对强度增益特性表

### 16.3.1 ISP\_TNR\_MANUAL\_ATTR\_S

#### 【说明】

TNR 属性手动参数

#### 【定义】

```
typedef struct _ISP_TNR_MANUAL_ATTR_S {
    CVI_U8 TnrStrength0;
    CVI_U8 MapThdLow0;
    CVI_U8 MapThdHigh0;
    CVI_U8 MtDetectUnit;
    CVI_S16 BrightnessNoiseLevelLE;
```

(下页继续)

(续上页)

```

CVI_S16 BrightnessNoiseLevelSE;
CVI_BOOL MtFiltMode;
CVI_U16 MtFiltWgt;
} ISP_TNR_MANUAL_ATTR_S;

```

**【成员】**

成员名称	描述
TnrStrength0	长曝光 TNR 强度增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MapThdLow0	长曝光 TNR 强度上限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MapThdHigh0	长曝光 TNR 强度下限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MtDetectUnit	运动侦测时的抗噪能力，数值越大，抗噪能力越强，但是侦测的细致度越小。 取值范围：[0x3, 0x6] 数据类型：CVI_U8
BrightnessNoiseLevelLE	长曝光亮度噪声容忍值 取值范围：[0x1, 0x3ff] 数据类型：CVI_S16
BrightnessNoiseLevelSE	短曝光亮度噪声容忍值 取值范围：[0x1, 0x3ff] 数据类型：CVI_S16
MtFiltMode	运动侦测滤波器模式 取值范围：[0x00, 0x01] 数据类型：CVI_BOOL
MtFiltWgt	运动侦测滤波器权重 取值范围：[0x1, 0x100] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRAttr

CVI\_ISP\_GetTNRAttr

## 16.3.2 ISP\_TNR\_AUTO\_ATTR\_S

### 【说明】

TNR 属性自动参数

### 【定义】

```
typedef struct _ISP_TNR_AUTO_ATTR_S {
    CVI_U8 TnrStrength0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 MapThdLow0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 MapThdHigh0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 MtDetectUnit[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_S16 BrightnessNoiseLevelLE[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_S16 BrightnessNoiseLevelSE[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_BOOL MtFiltMode[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 MtFiltWgt[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_AUTO_ATTR_S;
```

### 【成员】

成员名称	描述
TnrStrength0	长曝光 TNR 强度增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MapThdLow0	长曝光 TNR 强度上限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MapThdHigh0	长曝光 TNR 强度下限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MtDetectUnit	运动侦测时的抗噪能力，数值越大，抗噪能力越强，但是侦测的细致度越小。 取值范围：[0x3, 0x6] 数据类型：CVI_U8
BrightnessNoiseLevelLE	长曝光亮度噪声容忍值 取值范围：[0x1, 0x3ff] 数据类型：CVI_S16
BrightnessNoiseLevelSE	短曝光亮度噪声容忍值 取值范围：[0x1, 0x3ff] 数据类型：CVI_S16
MtFiltMode	运动侦测滤波器模式 取值范围：[0x00, 0x01] 数据类型：CVI_BOOL
MtFiltWgt	运动侦测滤波器权重 取值范围：[0x1, 0x100] 数据类型：CVI_U16

### 【注意事项】

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRAttr

CVI\_ISP\_GetTNRAttr

### 16.3.3 ISP\_TNR\_ATTR\_S

**【说明】**

TNR 属性参数

**【定义】**

```
typedef struct _ISP_TNR_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_BOOL TuningMode;  
    CVI_BOOL TnrMtMode;  
    CVI_BOOL YnrCnrSharpenMtMode;  
    CVI_BOOL PreSharpenMtMode;  
    CVI_U8 ChromaScalingDownMode;  
    CVI_BOOL CompGainEnable;  
    ISP_TNR_MANUAL_ATTR_S stManual;  
    ISP_TNR_AUTO_ATTR_S stAuto;  
} ISP_TNR_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	TNR 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
TuningMode	调整模式, 输出可视化辅助信息, 帮助用户调试 0: 不输出可视化辅助信息 1: 输出移动侦测可视化结果, 越亮代表运动越明显, 越暗则反之。 取值范围: [0, 1] 数据类型: CVI_BOOL
TnrMtMode	TNR Motion 模式: 0: Motion IIR(代表侦测的 motion 有经过 IIR 处理) 1: Motion history(代表侦测的 motion 没有经过 IIR 处理) 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
YnrCnrSharpenMtMode	YNR/CNR Motion 模式: 0: Motion IIR(代表侦测的 motion 有经过 IIR 处理) 1: Motion history(代表侦测的 motion 没有经过 IIR 处理) 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
PreSharpenMtMode	Presharpen Motion 模式: 0: Motion IIR(代表侦测的 motion 有经过 IIR 处理) 1: Motion history(代表侦测的 motion 没有经过 IIR 处理) 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
ChromaScalingDownMode	色度缩小模式
CompGainEnable	亮度补偿功能使能 取值范围: [0, 1] 数据类型: CVI_BOOL
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRAttr

CVI\_ISP\_GetTNRAttr

### 16.3.4 ISP\_TNR\_NOISE\_MODEL\_MANUAL\_ATTR\_S

**【说明】**

TNR Noise Model 属性手动参数

**【定义】**

```
typedef struct _ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S {  
    CVI_U8 RNoiseLevel0;  
    CVI_U8 GNoiseLevel0;  
    CVI_U8 BNoiseLevel0;  
    CVI_U8 RNoiseLevel1;  
    CVI_U8 GNoiseLevel1;  
    CVI_U8 BNoiseLevel1;  
    CVI_U8 RNoiseHiLevel0;  
    CVI_U8 GNoiseHiLevel0;  
    CVI_U8 BNoiseHiLevel0;  
    CVI_U8 RNoiseHiLevel1;  
    CVI_U8 GNoiseHiLevel1;  
    CVI_U8 BNoiseHiLevel1;  
} ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S;
```

**【成员】**



成员名称	描述
RNoiseLevel0	长曝光红色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseHiLevel0	长曝光红色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseLevel0	长曝光绿色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseHiLevel0	长曝光绿色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseLevel0	长曝光蓝色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseHiLevel0	长曝光蓝色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseLevel1	短曝光红色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseHiLevel1	短曝光红色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseLevel1	短曝光绿色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseHiLevel1	短曝光绿色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseLevel1	短曝光蓝色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseHiLevel1	短曝光蓝色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

[CVI\\_ISP\\_SetTNRNoiseModelAttr](#)

[CVI\\_ISP\\_GetTNRNoiseModelAttr](#)

### 16.3.5 ISP\_TNR\_NOISE\_MODEL\_AUTO\_ATTR\_S

**【说明】**

TNR Noise Model 属性自动参数

**【定义】**

```
typedef struct _ISP_TNR_NOISE_MODEL_AUTO_ATTR_S {  
    CVI_U8 RNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 GNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 RNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 GNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 RNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 GNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 RNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 GNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_TNR_NOISE_MODEL_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
RNoiseLevel0	长曝光红色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseHiLevel0	长曝光红色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseLevel0	长曝光绿色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseHiLevel0	长曝光绿色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseLevel0	长曝光蓝色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseHiLevel0	长曝光蓝色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseLevel1	短曝光红色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
RNoiseHiLevel1	短曝光红色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseLevel1	短曝光绿色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GNoiseHiLevel1	短曝光绿色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseLevel1	短曝光蓝色通道噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BNoiseHiLevel1	短曝光蓝色通道亮部噪声容忍值 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

[CVI\\_ISP\\_SetTNRNoiseModelAttr](#)

[CVI\\_ISP\\_GetTNRNoiseModelAttr](#)

### 16.3.6 ISP\_TNR\_NOISE\_MODEL\_ATTR\_S

**【说明】**

TNR Noise Model 属性参数

**【定义】**

```
typedef struct _ISP_TNR_NOISE_MODEL_ATTR_S {  
    ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S stManual;  
    ISP_TNR_NOISE_MODEL_AUTO_ATTR_S stAuto;  
} ISP_TNR_NOISE_MODEL_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRNoiseModelAttr

CVI\_ISP\_GetTNRNoiseModelAttr

### 16.3.7 ISP\_TNR\_LUMA\_MOTION\_MANUAL\_ATTR\_S

**【说明】**

手动模式 TNR 亮度对强度增益特性表

**【定义】**

```
typedef struct _ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S {  
    CVI_U16 L2mIn0[4];  
    CVI_U8 L2mOut0[4];  
    CVI_U16 L2mIn1[4];  
    CVI_U8 L2mOut1[4];  
    CVI_BOOL MtLumaMode;  
} ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
L2mIn0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
L2mOut0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
L2mIn1[4]	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
L2mOut1[4]	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
MtLumaMode	Luma gain 参考对象 0: luma 1: max(R,G,B) 取值范围：[0x0, 0x1] 数据类型：CVI_BOOL

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNR\_LumaMotionAttr

CVI\_ISP\_GetTNR\_LumaMotionAttr

## 16.3.8 ISP\_TNR\_LUMA\_MOTION\_AUTO\_ATTR\_S

**【说明】**

自动模式 TNR 亮度对强度增益特性表

**【定义】**

```
typedef struct _ISP_TNR_LUMA_MOTION_AUTO_ATTR_S {
    CVI_U16 L2mIn0[4][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 L2mOut0[4][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 L2mIn1[4][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 L2mOut1[4][ISP_AUTO_ISO_STRENGTH_NUM];
}
```

(下页继续)

(续上页)

```
CVI_BOOL MtLumaMode[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_LUMA_MOTION_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
L2mIn0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
L2mOut0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
L2mIn1[4]	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
L2mOut1[4]	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
MtLumaMode	亮度增益模式 取值范围：[0x0, 0x1] 数据类型：CVI_BOOL

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNR\_LumaMotionAttr

CVI\_ISP\_GetTNR\_LumaMotionAttr

## 16.3.9 ISP\_TNR\_LUMA\_MOTION\_ATTR\_S

**【说明】**

TNR 亮度对强度增益特性表

**【定义】**

```
typedef struct _ISP_TNR_LUMA_MOTION_ATTR_S {
    ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S stManual;
```

(下页继续)

(续上页)

```
ISP_TNR_LUMA_MOTION_AUTO_ATTR_S stAuto;
} ISP_TNR_LUMA_MOTION_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRlumaMotionAttr

CVI\_ISP\_GetTNRlumaMotionAttr

## 16.3.10 ISP\_TNR\_GHOST\_MANUAL\_ATTR\_S

**【说明】**

手动模式动量对拖尾消除程度特性表

**【定义】**

```
typedef struct _ISP_TNR_GHOST_MANUAL_ATTR_S {
    CVI_U8 PrvMotion0[4];
    CVI_U8 PrtctWgt0[4];
    CVI_U8 MotionHistoryStr;
} ISP_TNR_GHOST_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
PrvMotion0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PrtctWgt0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。 取值范围：[0x0, 0xf] 数据类型：CVI_U8
MotionHistoryStr	定义输出给 YNR 参考之移动轨迹保留程度。数值越高移动轨迹保留越长，反之移动轨迹保留越短 取值范围：[0x0, 0xf] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRGhostAttr

CVI\_ISP\_GetTNRGhostAttr

### 16.3.11 ISP\_TNR\_GHOST\_AUTO\_ATTR\_S

**【说明】**

自动模式动量对拖尾消除程度特性表

**【定义】**

```
typedef struct ISP_TNR_GHOST_AUTO_ATTR_S {  
    CVI_U8 PrvMotion0[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 PrtctWgt0[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 MotionHistoryStr[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_TNR_GHOST_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
PrvMotion0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。 定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PrtctWgt0[4]	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。 定义强度增益，值越大强度越强。 取值范围：[0x0, 0xf] 数据类型：CVI_U8
MotionHistoryStr	定义输出给 YNR 参考之移动轨迹保留程度。数值越高移动轨迹保留越长，反之移动轨迹保留越短 取值范围：[0x0, 0xf] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRGhostAttr

CVI\_ISP\_GetTNRGhostAttr



## 16.3.12 ISP\_TNR\_GHOST\_ATTR\_S

### 【说明】

动量对拖尾消除程度特性表

### 【定义】

```
typedef struct _ISP_TNR_GHOST_ATTR_S {  
    ISP_TNR_GHOST_MANUAL_ATTR_S stManual;  
    ISP_TNR_GHOST_AUTO_ATTR_S stAuto;  
} ISP_TNR_GHOST_ATTR_S;
```

### 【成员】

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

### 【注意事项】

无

### 【相关数据类型及接口】

CVI\_ISP\_SetTNRGhostAttr

CVI\_ISP\_GetTNRGhostAttr

## 16.3.13 ISP\_TNR\_MT\_PRT\_MANUAL\_ATTR\_S

### 【说明】

手动模式 TNR 动量保护属性参数

### 【定义】

```
typedef struct _ISP_TNR_MT_PRT_MANUAL_ATTR_S {  
    CVI_U8 LowMtPrtLevelY;  
    CVI_U8 LowMtPrtLevelU;  
    CVI_U8 LowMtPrtLevelV;  
    CVI_U8 LowMtPrtInY[4];  
    CVI_U8 LowMtPrtInU[4];  
    CVI_U8 LowMtPrtInV[4];  
    CVI_U8 LowMtPrtOutY[4];  
    CVI_U8 LowMtPrtOutU[4];  
    CVI_U8 LowMtPrtOutV[4];  
    CVI_U8 LowMtPrtAdvIn[4];  
    CVI_U8 LowMtPrtAdvOut[4];  
} ISP_TNR_MT_PRT_MANUAL_ATTR_S;
```

### 【成员】

成员名称	描述
LowMtPrtLevelY	Y 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtLevelU	U 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtLevelV	V 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInY[4]	定义 LUT 的横轴，亦即 Y 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInU[4]	定义 LUT 的横轴，亦即 U 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInV[4]	定义 LUT 的横轴，亦即 V 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutY[4]	定义 LUT 的纵轴，亦即 Y 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutU[4]	定义 LUT 的纵轴，亦即 U 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutV[4]	定义 LUT 的纵轴，亦即 V 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvIn[4]	定义 LUT 的横轴，亦即区块为单位的前后帧差值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvOut[4]	定义 LUT 的纵轴，亦即前后帧差值的保护增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRMtPrtAttr

CVI\_ISP\_GetTNRMtPrtAttr

### 16.3.14 ISP\_TNR\_MT\_PRT\_AUTO\_ATTR\_S

**【说明】**

自动模式 TNR 动量保护属性参数

**【定义】**

```
typedef struct _ISP_TNR_MT_PRT_AUTO_ATTR_S {  
    CVI_U8 LowMtPrtLevelY[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtLevelU[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtLevelV[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtInY[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtInU[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtInV[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtOutY[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtOutU[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtOutV[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtAdvIn[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LowMtPrtAdvOut[4][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_TNR_MT_PRT_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
LowMtPrtLevelY	Y 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtLevelU	U 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtLevelV	V 通道保护上限值。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInY[4]	定义 LUT 的横轴，亦即 Y 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInU[4]	定义 LUT 的横轴，亦即 U 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtInV[4]	定义 LUT 的横轴，亦即 V 通道 pixel 为单位的前后帧差值。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutY[4]	定义 LUT 的纵轴，亦即 Y 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutU[4]	定义 LUT 的纵轴，亦即 U 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtOutV[4]	定义 LUT 的纵轴，亦即 V 通道前后帧差值的保护程度。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvIn[4]	定义 LUT 的横轴，亦即区块为单位的前后帧差值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvOut[4]	定义 LUT 的纵轴，亦即前后帧差值的保护增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRMtPrtAttr

CVI\_ISP\_GetTNRMtPrtAttr

### 16.3.15 ISP\_TNR\_MT\_PRT\_ATTR\_S

**【说明】**

TNR 动量保护属性参数

**【定义】**

```
typedef struct _ISP_TNR_MT_PRT_ATTR_S {  
    CVI_BOOL LowMtPrtEn; /*RW; Range:[0, 1]*/  
    CVI_BOOL LowMtLowPassEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL LowMtPrtAdvLumaEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL LowMtPrtAdvMode; /*RW; Range:[0, 1]*/  
    CVI_U8 LowMtPrtAdvMax; /*RW; Range:[0, 0xff]*/  
    CVI_BOOL LowMtPrtAdvDebugMode; /*RW; Range:[0, 1]*/  
    CVI_U8 LowMtPrtAdvDebugIn[4]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 LowMtPrtAdvDebugOut[4]; /*RW; Range:[0x0, 0xff]*/  
    ISP_TNR_MT_PRT_MANUAL_ATTR_S stManual;  
    ISP_TNR_MT_PRT_AUTO_ATTR_S stAuto;  
} ISP_TNR_MT_PRT_ATTR_S;
```

**【成员】**

成员名称	描述
LowMtPrtEn	空域降噪微小运动保护使能。该功能可根据像素为单位的前后帧差值定义保护程度，保护程度越高越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0, 1] 数据类型：CVI_BOOL
LowMtLowPassEnable	弱运动侦测使用进行低通滤波 取值范围：[0, 1] 数据类型：CVI_BOOL
LowMtPrtAdvLumaEnable	空域降噪微小运动保护功能使能。该功能可根据区块为单位的前后帧差值定义保护程度，保护程度越高越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出 取值范围：[0, 1] 数据类型：CVI_BOOL
LowMtPrtAdvMode	空域降噪微小运动模式 取值范围：[0, 1] 数据类型：CVI_BOOL
LowMtPrtAdvMax	保护增益上限 取值范围：[0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvDebugMode	微小运动调整模式，输出可视化辅助信息，帮助用户调试 0: 输出移动侦测可视化结果。 1: 根据 LowMtPrtDebug LUT 输出移动侦测可视化结果 取值范围：[0, 1] 数据类型：CVI_BOOL
LowMtPrtAdvDebugIn[4]	调整模式，微小运动输入范围设定 取值范围：[0, 0xff] 数据类型：CVI_U8
LowMtPrtAdvDebugOut[4]	调整模式，微小运动输出数值设定 取值范围：[0, 0xff] 数据类型：CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetTNRMtPrtAttr

CVI\_ISP\_GetTNRMtPrtAttr

## 16.3.16 ISP\_TNR\_MOTION\_ADAPT\_MANUAL\_ATTR\_S

### 【说明】

手动模式动量对强度增益特性表

### 【定义】

```
typedef struct _ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S {  
    CVI_U8 AdaptNrLumaStrIn[4];  
    CVI_U8 AdaptNrLumaStrOut[4];  
    CVI_U8 AdaptNrChromaStrIn[4];  
    CVI_U8 AdaptNrChromaStrOut[4];  
} ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S;
```

### 【成员】

成员名称	描述
AdaptNrLumaStrIn	TNR 动量对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrLumaStrOut	TNR 动量对强度增益特性表。由四组数值组成的数组。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrChromaStrIn	TNR 动量对强度增益特性表。由四组数值组成的数组。定义色度等级，值越大色度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrChromaStrOut	TNR 动量对强度增益特性表。由四组数值组成的数组。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

### 【注意事项】

无

### 【相关数据类型及接口】

CVI\_ISP\_SetTNRMotionAdaptAttr

CVI\_ISP\_GetTNRMotionAdaptAttr

## 16.3.17 ISP\_TNR\_MOTION\_ADAPT\_AUTO\_ATTR\_S

### 【说明】

自动模式 TNR 动量保护属性参数

### 【定义】

```
typedef struct _ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S {  
    CVI_U8 AdaptNrLumaStrIn[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 AdaptNrLumaStrOut[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 AdaptNrChromaStrIn[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 AdaptNrChromaStrOut[4][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S;
```

### 【成员】

成员名称	描述
AdaptNrLumaStrIn	TNR 动量对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrLumaStrOut	TNR 动量对强度增益特性表。由四组数值组成的数组。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrChromaStrIn	TNR 动量对强度增益特性表。由四组数值组成的数组。定义色度等级，值越大色度越高。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
AdaptNrChromaStrOut	TNR 动量对强度增益特性表。由四组数值组成的数组。值越高时，受保护之运动区域会越倾向原像素值输出。反之，值越低越倾向时域降噪后的像素值输出。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

### 【注意事项】

无

### 【相关数据类型及接口】

CVI\_ISP\_SetTNRMotionAdaptAttr

CVI\_ISP\_GetTNRMotionAdaptAttr



### 16.3.18 ISP\_TNR\_MOTION\_ADAPT\_ATTR\_S

**【说明】**

TNR 动量保护属性参数

**【定义】**

```
typedef struct _ISP_TNR_MOTION_ADAP_ATTR_S {  
    ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S stManual;  
    ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S stAuto;  
} ISP_TNR_MOTION_ADAPT_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetTNRMotionAdaptAttr

CVI\_ISP\_GetTNRMotionAdaptAttr

# 17 Crosstalk

## 17.1 功能描述

可以平衡 raw 之间临近像素 Gr 和 Gb 之间的差异，从而有效防止 demosaic 插值算法产生的方格或其他类似 pattern。

## 17.2 API 参考

- CVI\_ISP\_SetCrosstalkAttr: 设置 Crosstalk 属性参数
- CVI\_ISP\_GetCrosstalkAttr: 获取 Crosstalk 属性参数

### 17.2.1 CVI\_ISP\_SetCrosstalkAttr

#### 【描述】

设置 Crosstalk 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCrosstalkAttr(VI_PIPE ViPipe, const ISP_CROSSTALK_ATTR_S_
→*pstCrosstalkAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCrosstalkAttr	Crosstalk 属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetCrosstalkAttr](#)

## 17.2.2 CVI\_ISP\_GetCrosstalkAttr

**【描述】**

获取 Crosstalk 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCrosstalkAttr(VI_PIPE ViPipe, ISP_CROSSTALK_ATTR_S_
→*pstCrosstalkAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCrosstalkAttr	Crosstalk 属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败, 其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetCrosstalkAttr](#)

## 17.3 数据类型

- `ISP_CROSSTALK_MANUAL_ATTR_S`: Crosstalk 属性手动参数
- `ISP_CROSSTALK_AUTO_ATTR_S`: Crosstalk 属性自动参数
- `ISP_CROSSTALK_ATTR_S`: Crosstalk 属性参数

### 17.3.1 `ISP_CROSSTALK_MANUAL_ATTR_S`

**【说明】**

Crosstalk 属性手动参数

**【定义】**

```
typedef struct _ISP_CROSSTALK_MANUAL_ATTR_S {  
    CVI_U16 Strength;  
} ISP_CROSSTALK_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
Strength	G 通道平衡全局强度 取值范围: [0x0, 0x100] 数据类型: CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

`CVI_ISP_SetCrosstalkAttr`

`CVI_ISP_GetCrosstalkAttr`

### 17.3.2 `ISP_CROSSTALK_AUTO_ATTR_S`

**【说明】**

Crosstalk 属性自动参数

**【定义】**

```
typedef struct _ISP_CROSSTALK_AUTO_ATTR_S {  
    CVI_U16 Strength[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CROSSTALK_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
Strength	G 通道平衡全局强度 取值范围：[0x0, 0x100] 数据类型：CVI_U16

【注意事项】

无

【相关数据类型及接口】

CVI\_ISP\_SetCrosstalkAttr

CVI\_ISP\_GetCrosstalkAttr

17.3.3 ISP\_CROSSTALK\_ATTR\_S

【说明】

Crosstalk 属性参数

【定义】

```
typedef struct _ISP_CROSSTALK_ATTR_S {
    CVI_BOOL Enable; /*RW; Range:[0, 1]*/
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval;
    CVI_U16 GrGbDiffThreSec[4]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 FlatThre[4]; /*RW; Range:[0x0, 0xffff]*/
    ISP_CROSSTALK_MANUAL_ATTR_S stManual;
    ISP_CROSSTALK_AUTO_ATTR_S stAuto;
} ISP_CROSSTALK_ATTR_S;
```

【成员】

成员名称	描述
Enable	GE 模块使能。0: 关闭。1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
GrGbDiffThreSec[4]	G 通道平衡节点 1-4 阈值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
FlatThre[4]	平坦区检测节点 1-4 阈值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetCrosstalkAttr

CVI\_ISP\_GetCrosstalkAttr

# 18 Demosaic

## 18.1 功能描述

把图像由 Bayer Pattern 内插出完整的 RGB 而提升细节与清晰度。

## 18.2 API 参考

- CVI\_ISP\_SetDemosaicAttr: 设置 Demosaic 属性参数
- CVI\_ISP\_GetDemosaicAttr: 获取 Demosaic 属性参数
- CVI\_ISP\_SetDemosaicDemoireAttr: 设置 Demosaic 去马赛克属性参数
- CVI\_ISP\_GetDemosaicDemoireAttr: 获取 Demosaic 去马赛克属性参数

### 18.2.1 CVI\_ISP\_SetDemosaicAttr

#### 【描述】

设置 Demosaic 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetDemosaicAttr(VI_PIPE ViPipe, const ISP_DEMOSAIC_ATTR_S_
↪ *pstDemosaicAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDemosaicAttr	Demosaic 属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetDemosaicAttr](#)

## 18.2.2 CVI\_ISP\_GetDemosaicAttr

**【描述】**

获取 Demosaic 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDemosaicAttr(VI_PIPE ViPipe, ISP_DEMOSAIC_ATTR_S_
↪ *pstDemosaicAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDemosaicAttr	Demosaic 属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**



CVI\_ISP\_SetDemosaicAttr

### 18.2.3 CVI\_ISP\_SetDemosaicDemoireAttr

#### 【描述】

设置 Demosaic 去马赛克属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetDemosaicDemoireAttr(VI_PIPE ViPipe, const ISP_DEMOSAIC_DEMOIRE_
→ATTR_S *pstDemosaicDemoireAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDemosaicDe- moireAttr	Demosaic 去马赛克属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

#### 【注意】

#### 【举例】

无

#### 【相关主题】

CVI\_ISP\_GetDemosaicDemoireAttr

### 18.2.4 CVI\_ISP\_GetDemosaicDemoireAttr

#### 【描述】

获取 Demosaic 去马赛克属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_GetDemosaicDemoireAttr(VI_PIPE ViPipe, ISP_DEMOSAIC_DEMOIRE_
→ATTR_S *pstDemosaicDemoireAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDemosaicDemoireAttr	Demosaic 去马赛克属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】****【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetDemosaicDemoireAttr](#)

## 18.3 数据类型

- `ISP_DEMOSAIC_MANUAL_ATTR_S`: Demosaic 属性手动参数
- `ISP_DEMOSAIC_AUTO_ATTR_S`: Demosaic 属性自动参数
- `ISP_DEMOSAIC_ATTR_S`: Demosaic 属性参数
- `ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S`: Demosaic 去马赛克属性手动参数
- `ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S`: Demosaic 去马赛克属性自动参数
- `ISP_DEMOSAIC_DEMOIRE_ATTR_S`: Demosaic 去马赛克属性参数
- `ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S`: Demosaic 滤波器属性手动参数
- `ISP_DEMOSAIC_FILTER_AUTO_ATTR_S`: Demosaic 滤波器属性自动参数
- `ISP_DEMOSAIC_FILTER_ATTR_S`: Demosaic 滤波器属性参数

### 18.3.1 ISP\_DEMOSAIC\_MANUAL\_ATTR\_S

**【说明】**

Demosaic 属性手动参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_MANUAL_ATTR_S {  
    CVI_U16 CoarseEdgeThr;  
    CVI_U16 CoarseStr;  
    CVI_U16 FineEdgeThr;  
    CVI_U16 FineStr;  
    CVI_U16 RbSigLumaThd;  
    CVI_U8 FilterMode;  
} ISP_DEMOSAIC_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
CoarseEdgeThr	边缘粗调侦测阈值。值越小，侦测为边缘的数量越多。建议搭配参数 CoarseStr 调试。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
CoarseStr	边缘粗调强度值。值越小，越偏方向性的处理。反之，越偏无方向性的处理。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FineEdgeThr	边缘细调侦测阈值。值越小，侦测为边缘的数量越多。建议搭配参数 FineStr 调试。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FineStr	边缘细调强度值。值越小，越偏方向性的处理。反之，越偏无方向性的处理。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FilterMode	DC 影像的锐化宽度，数值越大，边缘被锐利化的宽度越宽，视觉上会越醒目。 取值范围：[0x0, 0x1] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDemosaicAttr

CVI\_ISP\_GetDemosaicAttr

## 18.3.2 ISP\_DEMOSAIC\_AUTO\_ATTR\_S

### 【说明】

Demosaic 属性自动参数

### 【定义】

```
typedef struct _ISP_DEMOSAIC_AUTO_ATTR_S {  
    CVI_U16 CoarseEdgeThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 CoarseStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 FineEdgeThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 FineStr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 RbSigLumaThd[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 FilterMode[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_DEMOSAIC_AUTO_ATTR_S;
```

### 【成员】

成员名称	描述
CoarseEdgeThr	边缘粗调侦测阈值。值越小，侦测为边缘的数量越多。建议搭配参数 CoarseStr 调试。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
CoarseStr	边缘粗调强度值。值越小，越偏方向性的处理。反之，越偏无方向性的处理。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FineEdgeThr	边缘细调侦测阈值。值越小，侦测为边缘的数量越多。建议搭配参数 FineStr 调试。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FineStr	边缘细调强度值。值越小，越偏方向性的处理。反之，越偏无方向性的处理。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
FilterMode	DC 影像的锐化宽度，数值越大，边缘被锐利化的宽度越宽，视觉上会越醒目。 取值范围：[0x0, 0x1] 数据类型：CVI_U8

### 【注意事项】

无

### 【相关数据类型及接口】

CVI\_ISP\_SetDemosaicAttr

CVI\_ISP\_GetDemosaicAttr

### 18.3.3 ISP\_DEMOSAIC\_ATTR\_S

**【说明】**

Demosaic 属性参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_BOOL TuningMode;  
    CVI_BOOL RbVtEnable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_DEMOSAIC_MANUAL_ATTR_S stManual;  
    ISP_DEMOSAIC_AUTO_ATTR_S stAuto;  
} ISP_DEMOSAIC_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	Demosaic 模块使能。0: 关闭。1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
TuningMode	输出调试方案, 辅助调节参数。0: Demosaic 图像结果。1: 平坦/垂直/水平边缘侦测图像结果。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDemosaicAttr

CVI\_ISP\_GetDemosaicAttr

### 18.3.4 ISP\_DEMOSAIC\_DEMOIRE\_MANUAL\_ATTR\_S

#### 【说明】

Demosaic 去马赛克属性手动参数

#### 【定义】

```
typedef struct _ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S {
    CVI_U8 AntiFalseColorStr; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 SatGainIn[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 SatGainOut[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 ProtectColorGainIn[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 ProtectColorGainOut[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor1; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor2; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor3; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 EdgeGainIn[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 EdgeGainOut[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailGainIn[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailGainOut[2]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailDetectLumaStr; /*RW; Range:[0x0, 0xffff]*/
    CVI_U8 DetailSmoothStr; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 DetailWgtThr; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailWgtMin; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtMax; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtSlope; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 EdgeWgtNp; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 EdgeWgtThr; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 EdgeWgtMin; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtMax; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtSlope; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 DetailSmoothMapTh; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailSmoothMapMin; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailSmoothMapMax; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailSmoothMapSlope; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 LumaWgt; /*RW; Range:[0x0, 0xff]*/
} ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
DetailSmoothStr	细节平滑强度。值越大，平滑强度越强，对伪细节的抑制强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无

#### 【相关数据类型及接口】

CVI\_ISP\_SetDemosaicDemoireAttr

CVI\_ISP\_GetDemosaicDemoireAttr

### 18.3.5 ISP\_DEMOSAIC\_DEMOIRE\_AUTO\_ATTR\_S

#### 【说明】

Demosaic 去马赛克属性自动参数

#### 【定义】

```
typedef struct ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S {
    CVI_U8 AntiFalseColorStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 SatGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 SatGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 ProtectColorGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 ProtectColorGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor1[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor2[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 UserDefineProtectColor3[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 EdgeGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 EdgeGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DetailDetectLumaStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U8 DetailSmoothStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 DetailWgtThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailWgtMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 EdgeWgtNp[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 EdgeWgtThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 EdgeWgtMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 DetailSmoothMapTh[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailSmoothMapMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailSmoothMapMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailSmoothMapSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 LumaWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
}ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S;
```

#### 【成员】

成员名称	描述
DetailSmoothStr	细节平滑强度。值越大，平滑强度越强，对伪细节的抑制强度越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDemosaicDemoireAttr

CVI\_ISP\_GetDemosaicDemoireAttr

## 18.3.6 ISP\_DEMOSAIC\_DEMOIRE\_ATTR\_S

**【说明】**

Demosaic 去马赛克属性参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_DEMOIRE_ATTR_S {  
    CVI_BOOL AntiFalseColorEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL ProtectColorEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL DetailDetectLumaEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL DetailSmoothEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL DetailMode; /*RW; Range:[0, 1]*/  
    ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S stManual;  
    ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S stAuto;  
} ISP_DEMOSAIC_DEMOIRE_ATTR_S;
```

**【成员】**

成员名称	描述
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDemosaicDemoireAttr

CVI\_ISP\_GetDemosaicDemoireAttr

## 18.3.7 ISP\_DEMOSAIC\_FILTER\_MANUAL\_ATTR\_S

**【说明】**

Demosaic 滤波器属性手动参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S {  
    CVI_U16 DetailWgtThr;  
    CVI_U8 DetailWgtSlope;
```

(下页继续)



(续上页)

```
CVI_U8 DetailWgtMin;  
CVI_U8 DetailWgtMax;  
CVI_U16 EdgeWgtThr;  
CVI_U8 EdgeWgtSlope;  
CVI_U8 EdgeWgtMin;  
CVI_U8 EdgeWgtMax;  
CVI_U16 LumaWgtThr;  
CVI_U8 LumaWgtSlope;  
CVI_U8 LumaWgtMin;  
CVI_U8 LumaWgtMax;  
} ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
DetailWgtThr	细节保留范围阈值。值越小，细节保留作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DetailWgtSlope	细节保留强度。值越大，细节保留越多。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DetailWgtMin	边缘细节平滑保留允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DetailWgtMax	边缘细节平滑保留允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtThr	细节保留范围阈值。值越小，细节保留作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
EdgeWgtSlope	细节保留强度。值越大，细节保留越多。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtMin	边缘细节平滑保留允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtMax	边缘细节平滑保留允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtThr	亮度细节平滑作用范围阈值。值越小，细节平滑作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
LumaWgtSlope	亮度细节平滑边缘保留强度。值越大，细节平滑强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMin	亮度细节平滑强度允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMax	亮度细节平滑强度允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

cv180x & cv181x 不支持此功能

**【相关数据类型及接口】** 无

### 18.3.8 ISP\_DEMOSAIC\_FILTER\_AUTO\_ATTR\_S

**【说明】**

Demosaic 滤波器属性自动参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_FILTER_AUTO_ATTR_S {  
    CVI_U16 DetailWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DetailWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DetailWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DetailWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 EdgeWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 EdgeWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 EdgeWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 EdgeWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 LumaWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_DEMOSAIC_FILTER_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
DetailWgtThr	细节保留范围阈值。值越小，细节保留作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DetailWgtSlope	细节保留强度。值越大，细节保留越多。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DetailWgtMin	边缘细节平滑保留允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DetailWgtMax	边缘细节平滑保留允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtThr	细节保留范围阈值。值越小，细节保留作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
EdgeWgtSlope	细节保留强度。值越大，细节保留越多。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtMin	边缘细节平滑保留允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeWgtMax	边缘细节平滑保留允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtThr	亮度细节平滑作用范围阈值。值越小，细节平滑作用的范围越大。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
LumaWgtSlope	亮度细节平滑边缘保留强度。值越大，细节平滑强度越强。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMin	亮度细节平滑强度允许之最小增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMax	亮度细节平滑强度允许之最大增益。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

cv180x & cv181x 不支持此功能

**【相关数据类型及接口】**

无

### 18.3.9 ISP\_DEMOSAIC\_FILTER\_ATTR\_S

**【说明】**

Demosaic 滤波器属性参数

**【定义】**

```
typedef struct _ISP_DEMOSAIC_FILTER_ATTR_S {  
    ISP_OP_TYPE_E enOpType;  
    ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S stManual;  
    ISP_DEMOSAIC_FILTER_AUTO_ATTR_S stAuto;  
} ISP_DEMOSAIC_FILTER_ATTR_S;
```

**【成员】**

成员名称	描述
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

cv180x & cv181x 不支持此功能

**【相关数据类型及接口】**

无

# 19 Sharpen

## 19.1 功能描述

此模块用于增强图像清晰度，位于 3DNR 之后，主要锐化图像中的大边缘。

## 19.2 API 参考

- CVI\_ISP\_SetSharpenAttr: 设置锐化属性参数
- CVI\_ISP\_GetSharpenAttr: 获取锐化属性参数

### 19.2.1 CVI\_ISP\_SetSharpenAttr

【描述】

设置锐化属性参数

【语法】

```
CVI_S32 CVI_ISP_SetSharpenAttr(VI_PIPE ViPipe, const ISP_SHARPEN_ATTR_S *pstSharpenAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstSharpenAttr	锐化属性参数	输入

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetSharpenAttr](#)

## 19.2.2 CVI\_ISP\_GetSharpenAttr

**【描述】**

获取锐化属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetSharpenAttr(VI_PIPE ViPipe, ISP_SHARPEN_ATTR_S * pstSharpenAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstSharpenAttr	锐化属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetSharpenAttr](#)

## 19.3 数据类型

- ISP\_SHARPEN\_MANUAL\_ATTR\_S: 锐化属性手动参数
- ISP\_SHARPEN\_AUTO\_ATTR\_S: 锐化属性自动参数
- ISP\_SHARPEN\_ATTR\_S: 锐化属性参数

### 19.3.1 ISP\_SHARPEN\_MANUAL\_ATTR\_S

#### 【说明】

锐化属性手动参数

#### 【定义】

```
typedef struct _ISP_SHARPEN_MANUAL_ATTR_S {  
    CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/  
    CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM];  
    CVI_U8 LumaCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 LumaCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/  
    CVI_U8 MotionCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MotionCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/  
    CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/  
    CVI_U8 GlobalGain; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/  
    CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/  
    CVI_U8 HFBlendWgt; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MFBlendWgt; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 OverShootThr; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 UnderShootThr; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 OverShootThrMax; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 UnderShootThrMin; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MotionShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MotionShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/  
    CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/  
    CVI_U8 SatShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 SatShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/  
} ISP_SHARPEN_MANUAL_ATTR_S;
```

#### 【成员】



成员名称	描述
LumaAdpGain[33]	亮度锐化权重 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
LumaCor-LutIn[EE_LUT_NODE]	基于 luma 的 coring，此为输入节点，输入 luma 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaCor-LutOut[EE_LUT_NODE]	基于 luma 的 coring，此为输出节点，输出对应 coring 值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCor-LutIn[EE_LUT_NODE]	基于 motion 的 coring，此为输入节点，输入为 motion 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCor-LutOut[EE_LUT_NODE]	基于 motion 的 coring，此为输出节点，输出对应 coring 值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorWgt-LutIn[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重，此为输入节点，输入 motion 的大小 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorWgt-LutOut[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重，此为输出节点，输出 motion coring 的权重。(max = 128) 取值范围：[0x0, 0x80] 数据类型：CVI_U8
GlobalGain	全局锐化权重 取值范围：[0x0, 0xff] 数据类型：CVI_U8
OverShootGain	白边锐化权重 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
UnderShootGain	黑边锐化权重 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
EdgeGlobalGain	全局边缘锐化权重 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
DeltaShtCtrl[33]	根据边缘侦测的结果定义边缘增强的程度 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
DeltaShtCtrlUnit	边缘侦测的强度调控 取值范围：[0x0, 0x3] 数据类型：CVI_U8
YNoiseLevel	亮度锐化噪声值放大倍数，一倍为 64。数值越大亮度锐化噪声越被放大，反之则缩小 取值范围：[0x0, 0xff] 数据类型：CVI_U8
HFBlendWgt	高频边缘加强的权重 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MFBlendWgt	中频边缘加强的权重 取值范围：[0x0, 0xff] 数据类型：CVI_U8
OverShootThr	白边锐化上限幅度 取值范围：[0x0, 0xff]

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetSharpenAttr

CVI\_ISP\_GetSharpenAttr

**19.3.2 ISP\_SHARPEN\_AUTO\_ATTR\_S****【说明】**

锐化属性自动参数

**【定义】**

```
typedef struct _ISP_SHARPEN_MANUAL_ATTR_S {
    CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x3f]*/
    CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 LumaCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 LumaCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x20]*/
    CVI_U8 MotionCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x20]*/
    CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
    CVI_U8 GlobalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 UnderShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 HFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThrMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThrMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
    CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x3f]*/
    CVI_U8 SatShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 SatShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
} ISP_SHARPEN_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
LumaAdpGain[33]	亮度锐化权重 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
LumaCorLutIn[EE_LUT_NODE]	基于 luma 的 coring, 此为输入节点, 输入 luma 取值范围: [0x0, 0xff] 数据类型: CVI_U8
LumaCorLutOut[EE_LUT_NODE]	基于 luma 的 coring, 此为输出节点, 输出对应 coring 值 取值范围: [0x0, 0xff] 数据类型: CVI_U8
MotionCorLutIn[EE_LUT_NODE]	基于 motion 的 coring, 此为输入节点, 输入为 motion 取值范围: [0x0, 0xff] 数据类型: CVI_U8
MotionCorLutOut[EE_LUT_NODE]	基于 motion 的 coring, 此为输出节点, 输出对应 coring 值 取值范围: [0x0, 0xff] 数据类型: CVI_U8
MotionCorWgtLutIn[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重, 此为输入节点, 输入 motion 的大小 取值范围: [0x0, 0xff] 数据类型: CVI_U8
MotionCorWgtLutOut[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重, 此为输出节点, 输出 motion coring 的权重。(max = 128) 取值范围: [0x0, 0x80] 数据类型: CVI_U8
GlobalGain	全局锐化权重 取值范围: [0x0, 0xff] 数据类型: CVI_U8
OverShootGain	白边锐化权重 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
UnderShootGain	黑边锐化权重 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
EdgeGlobalGain	全局边缘锐化权重 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
DeltaShtCtrl[33]	根据边缘侦测的结果定义边缘增强的程度 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
DeltaShtCtrlUnit	边缘侦测的强度调控 取值范围: [0x0, 0x3] 数据类型: CVI_U8
YNoiseLevel	亮度锐化噪声值放大倍率, 一倍为 64。数值越大亮度锐化噪声越被放大, 反之则缩小 取值范围: [0x0, 0xff] 数据类型: CVI_U8
HFBlendWgt	高频边缘加强的权重 取值范围: [0x0, 0xff]

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetSharpenAttr

CVI\_ISP\_GetSharpenAttr

### 19.3.3 ISP\_SHARPEN\_ATTR\_S

**【说明】**

锐化属性参数

**【定义】**

```
typedef struct ISP_SHARPEN_ATTR_S {  
    CVI_BOOL Enable; /*RW; Range:[0x0, 0x1]*/  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/  
    CVI_U8 TuningMode; /*RW; Range:[0x0, 0xb]*/  
    CVI_BOOL LumaAdpGainEn; /*RW; Range:[0x0, 0x1]*/  
    CVI_BOOL DeltaAdpGainEn; /*RW; Range:[0x0, 0x1]*/  
    CVI_BOOL NoiseSuppressEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL SatShtCtrlEn; /*RW; Range:[0, 1]*/  
    CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/  
    CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/  
    ISP_SHARPEN_MANUAL_ATTR_S stManual;  
    ISP_SHARPEN_AUTO_ATTR_S stAuto;  
} ISP_SHARPEN_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	Y Sharpen 模块使能 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好 取值范围: [0x0, 0xff] 数据类型: CVI_U8
TuningMode	输出调试方案, 辅助调节参数 取值范围: [0x0, 0xb] 数据类型: CVI_U8
LumaAdpCoringEn	自动亮度噪声抑止阈值开关 取值范围: [0, 1] 数据类型: CVI_BOOL
LumaAdpGainEn	亮度锐化权重使能 取值范围: [0, 1] 数据类型: CVI_BOOL
DeltaAdpGainEn	锐化权重使能 取值范围: [0, 1] 数据类型: CVI_BOOL
DeltaAdpGain[SHARPEN_LUT_NUM]	锐化权重 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
NoiseSuppressEnable	加强去噪使能。针对边缘检测的影像做加强去噪的前处理后再做边缘增强 取值范围: [0, 1] 数据类型: CVI_BOOL
SatShtCtrlEn	由饱和度调整边缘增强的使能 取值范围: [0, 1] 数据类型: CVI_BOOL
SoftClampEnable	平滑的处理边缘加强 取值范围: [0, 1] 数据类型: CVI_BOOL
SoftClampUB	平滑处理边缘加强的上界, 设定的值越大, 则边缘加强的越连续, 但加强的效果也越弱 取值范围: [0, 0xff] 数据类型: CVI_U8
SoftClampLB	平滑处理边缘加强的下界, 设定的值越大, 则边缘加强的越连续, 但加强的效果也越弱 取值范围: [0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

## 【注意事项】

无

**【相关数据类型及接口】**

CVI\_ISP\_SetSharpenAttr

CVI\_ISP\_GetSharpenAttr

# 20 PreSharpen

---

## 20.1 功能描述

此模块用于增强图像清晰度，位于 3DNR 之前，主要锐化图像中具方向性的边缘和细节纹理。

## 20.2 API 参考

- CVI\_ISP\_SetPreSharpenAttr: 设置锐化属性参数
- CVI\_ISP\_GetPreSharpenAttr: 获取锐化属性参数

### 20.2.1 CVI\_ISP\_SetPreSharpenAttr

#### 【描述】

设置锐化属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetPreSharpenAttr(VI_PIPE ViPipe, const ISP_PRESHARPEN_ATTR_S *  
↪pstPreSharpenAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstPreSharpenAttr	锐化属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无

【举例】

无

【相关主题】

[CVI\\_ISP\\_GetPreSharpenAttr](#)

### 20.2.2 CVI\_ISP\_GetPreSharpenAttr

【描述】

获取锐化属性参数

【语法】

```
CVI_S32 CVI_ISP_GetPreSharpenAttr(VI_PIPE ViPipe, ISP_PRESHARPEN_ATTR_S *  
    ↪pstPreSharpenAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstPreSharpenAttr	锐化属性参数	输出

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无

【举例】

无

【相关主题】

[CVI\\_ISP\\_SetPreSharpenAttr](#)



## 20.3 数据类型

- ISP\_PRESHARPEN\_MANUAL\_ATTR\_S: 锐化属性手动参数
- ISP\_PRESHARPEN\_AUTO\_ATTR\_S: 锐化属性自动参数
- ISP\_PRESHARPEN\_ATTR\_S: 锐化属性参数

### 20.3.1 ISP\_PRESHARPEN\_MANUAL\_ATTR\_S

#### 【说明】

锐化属性手动参数

#### 【定义】

```
typedef struct _ISP_PRESHARPEN_MANUAL_ATTR_S {
    CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 LumaCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 LumaCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
    CVI_U8 MotionCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
    CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
    CVI_U8 GlobalGain; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 HFBlendWgt; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MFBlendWgt; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThr; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThr; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThrMax; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThrMin; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
    CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 SatShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 SatShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
} ISP_PRESHARPEN_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
EdgeGain	边缘的增强参数 值越大时，边缘锐化强度越大 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
TextureGain	细节纹理的增强参数 值越大时，细节纹理锐化强度越大 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
EdgeFreq	图像具方向性的边缘频段控制 取值范围：[0x0, 0x4] 数据类型：CVI_U8
TextureFreq	图像无方向性的细节纹理频段控制 取值范围：[0x0, 0x4] 数据类型：CVI_U8
LumaAdpCoring[33]	亮度锐化噪声值，细节纹理或边缘的增强会排除该容忍值所贡献的增强。数值越小锐化的效果越明显但也越容易将噪声锐化。数值越大则不易将噪声锐化，但锐化的效果则越不明显 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
LumaAdpGain[33]	亮度锐化权重 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
LumaCorLutIn[EE_LUT_NODE]	基于 luma 的 coring，此为输入节点，输入 luma 取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaCorLutOut[EE_LUT_NODE]	基于 luma 的 coring，此为输出节点，输出对应 coring 值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorLutIn[EE_LUT_NODE]	基于 motion 的 coring，此为输入节点，输入为 motion 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorLutOut[EE_LUT_NODE]	基于 motion 的 coring，此为输出节点，输出对应 coring 值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorWgtLutIn[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重，此为输入节点，输入 motion 的大小 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionCorWgtLutOut[EE_LUT_NODE]	基于 motion 调整 luma coring 以及 motion coring 的权重，此为输出节点，输出 motion coring 的权重。(max = 128) 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GlobalGain	全局锐化权重 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeGlobalGain	全局边缘锐化权重 取值范围：[0x0, 0x3f]

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetPreSharpenAttr

CVI\_ISP\_GetPreSharpenAttr

## 20.3.2 ISP\_PRESHARPEN\_AUTO\_ATTR\_S

## 【说明】

锐化属性自动参数

## 【定义】

```
typedef struct _ISP_PRESHARPEN_AUTO_ATTR_S {
    CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x3f]*/
    CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x3f]*/
    CVI_U8 LumaCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 LumaCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x20]*/
    CVI_U8 MotionCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x20]*/
    CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
    CVI_U8 GlobalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 UnderShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 HFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootThrMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThrMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 MotionShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
    CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x3f]*/
    CVI_U8 SatShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0xff]*/
    CVI_U8 SatShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;↵
↵Range:[0x0, 0x80]*/
} ISP_PRESHARPEN_AUTO_ATTR_S;
```

## 【成员】

成员名称	描述
EdgeGain	边缘的增强参数 值越大时，边缘锐化强度越大 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
OverShootThr	白边锐化上限幅度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
UnderShootThr	黑边锐化下限幅度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
YNoiseLevel	亮度锐化噪声值放大倍率，一倍为 64。数值越大亮度锐化噪声越被放大，反之则缩小 取值范围：[0x0, 0xff] 数据类型：CVI_U8
OverShootThrMax	白边锐化最大上限幅度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
UnderShootThrMin	黑边锐化最大下限幅度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionShtGainIn[4]	针对 motion 区决定边缘增强程度的 LUT，此为水平节点，输入值为 motion 值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionShtGainOut[4]	针对 motion 区决定边缘增强程度的 LUT，此为垂直节点，输出值为 motion 所对应的增强强度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
HueShtCtrl[SHARPEN_LUT_NUM]	基于指定的色彩做边缘增强 取值范围：[0x0, 0x3f] 数据类型：CVI_U8
SatShtGainIn[EE_LUT_NODE]	基于指定的饱和度做边缘增强，此为输入节点，输入饱和度 取值范围：[0x0, 0xff] 数据类型：CVI_U8
SatShtGainOut[EE_LUT_NODE]	基于指定的饱和度做边缘增强，此为输出节点，输出对应饱和度的边缘强度。(max = 128) 取值范围：[0x0, 0xff] 数据类型：CVI_U8

## 【注意事项】

无

## 【相关数据类型及接口】

CVI\_ISP\_SetPreSharpenAttr

CVI\_ISP\_GetPreSharpenAttr

### 20.3.3 ISP\_PRESHARPEN\_ATTR\_S

#### 【说明】

锐化属性参数

#### 【定义】

```
typedef struct _ISP_PRESHARPEN_ATTR_S {  
    CVI_BOOL Enable; /*RW; Range:[0x0, 0x1]*/  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/  
    CVI_U8 TuningMode; /*RW; Range:[0x0, 0xb]*/  
    CVI_BOOL LumaAdpGainEn; /*RW; Range:[0x0, 0x1]*/  
    CVI_BOOL DeltaAdpGainEn; /*RW; Range:[0x0, 0x1]*/  
    CVI_BOOL NoiseSuppressEnable; /*RW; Range:[0, 1]*/  
    CVI_BOOL SatShtCtrlEn; /*RW; Range:[0, 1]*/  
    CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/  
    CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/  
    ISP_PRESHARPEN_MANUAL_ATTR_S stManual;  
    ISP_PRESHARPEN_AUTO_ATTR_S stAuto;  
} ISP_PRESHARPEN_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	Y Sharpen 模块使能 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
TuningMode	输出调试方案, 辅助调节参数 取值范围: [0x0, 0xb] 数据类型: CVI_U8
TuningModeLevelShift	针对 TuningMode 1, 2, 及 3 的辅助调节参数增益 取值范围: [0x0, 0x8] 数据类型: CVI_U8
FilterCoefByISOFuncEn	动态滤波系数使能 取值范围: [0, 1] 数据类型: CVI_BOOL
FilterCoefByISOLowThd	动态滤波系数下阈值, 低于此值偏好使用有向性滤波系数 取值范围: [0, 0xffffffff] 数据类型: CVI_U32
FilterCoefByISOHighThd	动态滤波系数上阈值, 低于此值偏好使用无向性滤波系数 取值范围: [0, 0xffffffff] 数据类型: CVI_U32
FilterCoefNonDirType	动态滤波系数类别 取值范围: [0, 0xff] 数据类型: CVI_U32
LumaAdpCoringEn	自动亮度噪声抑止阈值开关 取值范围: [0, 1] 数据类型: CVI_BOOL
LumaAdpGainEn	亮度锐化权重使能 取值范围: [0, 1] 数据类型: CVI_BOOL
DeltaAdpGainEn	锐化权重使能 取值范围: [0, 1] 数据类型: CVI_BOOL
WdrCoringCompensationEn	亮度锐化噪声值在 wdr 模式的补偿使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
WdrCoringCompensationMode	亮度锐化噪声值在 wdr 模式的补偿模式。 0: 根据 WdrCoringHighThrd 与 WdrCoringLowthd 补偿亮度锐化噪声。1: 根据 DRC tone mapping curve 自动补偿亮度锐化噪声。 取值范围: [0, 1] 数据类型: CVI_BOOL
WdrCoringToleranceLevel	亮度锐化噪声值在 wdr 模式补偿的容忍值。数值越小锐化的效果越明显但也越容易将噪声锐化。数值越大则不易将噪声锐化, 但锐化的效果则越不明显。

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetPreSharpenAttr

CVI\_ISP\_GetPreSharpenAttr

# 21 Gamma

## 21.1 功能描述

因人眼对低亮度较敏感, 故对图像的亮度做非线性转换, 压缩亮区、拉伸暗区使整体亮度感受更好。

## 21.2 API 参考

- CVI\_ISP\_SetGammaAttr: 设置 gamma 属性参数
- CVI\_ISP\_GetGammaAttr: 获取 gamma 属性参数
- CVI\_ISP\_GetGammaCurveByType: 获取 Gamma 曲线信息
- CVI\_ISP\_SetAutoGammaAttr: 设置 Auto Gamma 属性参数
- CVI\_ISP\_GetAutoGammaAttr: 获取 Auto Gamma 属性参数

### 21.2.1 CVI\_ISP\_SetGammaAttr

#### 【描述】

设置 gamma 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetGammaAttr(VI_PIPE ViPipe, const ISP_GAMMA_ATTR_S_
↪*pstGammaAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstGammaAttr	gamma 属性参数	输入

#### 【返回值】



返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetGammaAttr](#)

## 21.2.2 CVI\_ISP\_GetGammaAttr

**【描述】**

获取 gamma 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetGammaAttr(VI_PIPE ViPipe, ISP_GAMMA_ATTR_S *pstGammaAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstGammaAttr	gamma 属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetGammaAttr](#)

## 21.2.3 CVI\_ISP\_GetGammaCurveByType

**【描述】**

获取 Gamma 曲线信息

**【语法】**

```
CVI_S32 CVI_ISP_GetGammaCurveByType(VI_PIPE ViPipe, ISP_GAMMA_ATTR_S_
↪ *pstGammaAttr, const ISP_GAMMA_CURVE_TYPE_E curveType);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstGammaAttr	Gamma 属性参数	输出
curveType	Gamma 曲线类型	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetGammaAttr](#)

[CVI\\_ISP\\_GetGammaAttr](#)

## 21.2.4 CVI\_ISP\_SetAutoGammaAttr

### 【描述】

设置 Auto Gamma 属性参数

### 【语法】

```
CVI_S32 CVI_ISP_SetAutoGammaAttr(VI_PIPE ViPipe, const ISP_AUTO_GAMMA_ATTR_S_
→*pstGammaAttr);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstGammaAttr	Auto Gamma 属性参数	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无

### 【举例】

无

### 【相关主题】

[CVI\\_ISP\\_GetAutoGammaAttr](#)

## 21.2.5 CVI\_ISP\_GetAutoGammaAttr

### 【描述】

获取 Auto Gamma 属性参数

### 【语法】

```
CVI_S32 CVI_ISP_GetAutoGammaAttr(VI_PIPE ViPipe, ISP_AUTO_GAMMA_ATTR_S_
→*pstGammaAttr);
```

## 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstGammaAttr	Auto Gamma 属性参数	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

## 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

## 【注意】

无

## 【举例】

无

## 【相关主题】

`CVI_ISP_SetAutoGammaAttr`

## 21.3 数据类型

- `ISP_GAMMA_ATTR_S`: Gamma 属性参数
- `ISP_GAMMA_CURVE_ATTR_S`: Auto Gamma 曲线参数
- `ISP_AUTO_GAMMA_ATTR_S`: Auto Gamma 属性参数

### 21.3.1 ISP\_GAMMA\_CURVE\_TYPE\_E

## 【说明】

Gamma 曲线属性

## 【定义】

```
typedef enum ISP_GAMMA_CURVE_TYPE_E {  
    ISP_GAMMA_CURVE_DEFAULT,  
    ISP_GAMMA_CURVE_SRGB,  
    ISP_GAMMA_CURVE_USER_DEFINE,  
    ISP_GAMMA_CURVE_AUTO,  
}
```

(下页继续)

(续上页)

```
ISP_GAMMA_CURVE_MAX  
} ISP_GAMMA_CURVE_TYPE_E;
```

**【成员】**

成员名称	描述
ISP_GAMMA_CURVE_DEFAULT	默认曲线
ISP_GAMMA_CURVE_SRGB	SRGB 曲线
ISP_GAMMA_CURVE_USER_DEFINE	用户自定义曲线
ISP_GAMMA_CURVE_AUTO	自适应曲线请参考 <a href="#">CVI_ISP_SetAutoGammaAttr</a> <a href="#">CVI_ISP_GetAutoGammaAttr</a>
ISP_GAMMA_CURVE_MAX	

**【注意事项】**

无

**【相关数据类型及接口】**

[CVI\\_ISP\\_SetGammaAttr](#)

[CVI\\_ISP\\_GetGammaAttr](#)

[CVI\\_ISP\\_GetGammaCurveByType](#)

## 21.3.2 ISP\_GAMMA\_ATTR\_S

**【说明】**

gamma 属性参数

**【定义】**

```
typedef struct _ISP_GAMMA_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_U8 UpdateInterval;  
    CVI_U16 Table[GAMMA_NODE_NUM];  
    ISP_GAMMA_CURVE_TYPE_E enCurveType;  
} ISP_GAMMA_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	Gamma 功能使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
Table[GAMMA_NODE_NUM]	Gamma 曲线节点数值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
enCurveType	Gamma 曲线类型

**【注意事项】**

GAMMA\_NODE\_NUM = 256

**【相关数据类型及接口】**

CVI\_ISP\_SetGammaAttr

CVI\_ISP\_GetGammaAttr

CVI\_ISP\_GetGammaCurveByType

### 21.3.3 ISP\_GAMMA\_CURVE\_ATTR\_S

**【说明】**

gamma 属性参数

**【定义】**

```
typedef struct _ISP_GAMMA_CURVE_ATTR_S {  
    CVI_S16 Lv;  
    CVI_U16 Tbl[GAMMA_NODE_NUM];  
} ISP_GAMMA_CURVE_ATTR_S;
```

**【成员】**

成员名称	描述
Lv	Gamma 曲线对应 Lv 范围 取值范围: [-500, 1500] 数据类型: CVI_S16
Tbl[GAMMA_NODE_NUM]	Gamma 曲线节点数值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16

**【注意事项】**

GAMMA\_NODE\_NUM = 256

【相关数据类型及接口】

CVI\_ISP\_SetAutoGammaAttr

CVI\_ISP\_GetAutoGammaAttr

21.3.4 ISP\_AUTO\_GAMMA\_ATTR\_S

【说明】

gamma 属性参数

【定义】

```
typedef struct _ISP_AUTO_GAMMA_ATTR_S {
    CVI_U8 GammaTabNum;
    ISP_GAMMA_CURVE_ATTR_S GammaTab[GAMMA_MAX_INTERPOLATION_NUM];
} ISP_AUTO_GAMMA_ATTR_S;
```

【成员】

成员名称	描述
GammaTabNum	自适应 Gamma 曲线数量 取值范围：[0, 5] 数据类型：CVI_U8
GammaTab[GAMMA_MAX_INTERPOLATION_NUM]	自适应 Gamma 曲线信息

【注意事项】

GAMMA\_MAX\_INTERPOLATION\_NUM=5

【相关数据类型及接口】

CVI\_ISP\_SetAutoGammaAttr

CVI\_ISP\_GetAutoGammaAttr

# 22 DCI

## 22.1 功能描述

Dynamic Contrast Improvement 的简称，藉由调整直方图来加强图像对比度，提升暗部细节。

## 22.2 API 参考

- `CVI_ISP_SetDCIAttr`: 设置 DCI 参数属性
- `CVI_ISP_GetDCIAttr`: 获取 DCI 参数属性

### 22.2.1 CVI\_ISP\_SetDCIAttr

#### 【描述】

设置 DCI 参数属性

#### 【语法】

```
CVI_S32 CVI_ISP_SetDCIAttr(VI_PIPE ViPipe, const ISP_DCI_ATTR_S *pstDCIAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDCIAttr	DCI 参数属性	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`



· 库文件: libisp.so

【注意】

无

【举例】

无

【相关主题】

[CVI\\_ISP\\_GetDCIAttr](#)

22.2.2 CVI\_ISP\_GetDCIAttr

【描述】

获取 DCI 参数属性

【语法】

```
CVI_S32 CVI_ISP_GetDCIAttr(VI_PIPE ViPipe, ISP_DCI_ATTR_S *pstDCIAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDCIAttr	DCI 参数属性	输出

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

【注意】

无

【举例】

无

【相关主题】

[CVI\\_ISP\\_SetDCIAttr](#)

## 22.3 数据类型

- `ISP_DCI_MANUAL_ATTR_S`: DCI 参数手动属性
- `ISP_DCI_AUTO_ATTR_S`: DCI 参数自动属性
- `ISP_DCI_ATTR_S`: DCI 参数属性

### 22.3.1 `ISP_DCI_MANUAL_ATTR_S`

#### 【说明】

DCI 参数手动属性

#### 【定义】

```
typedef struct _ISP_DCI_MANUAL_ATTR_S {  
    CVI_U16 ContrastGain;  
    CVI_U8 BlcThr;  
    CVI_U8 WhtThr;  
    CVI_U16 BlcCtrl;  
    CVI_U16 WhtCtrl;  
    CVI_U16 DciGainMax;  
} ISP_DCI_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
ContrastGain	用来控制 DCI 的强度，值越大，对比度越大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
BlcThr	用来决定暗区范围的门阈值。值越大，包含的暗区范围越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
WhtThr	用来决定亮区范围的门阈值。值越小，包含的亮区范围越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BlcCtrl	用来决定暗区的对比度。数值为 256 时，暗区对比度不变。比 256 大时，值越大，暗区对比度越大；反之，比 256 小时，值越小，暗区对比度越小。 取值范围：[0x0, 0x200] 数据类型：CVI_U16
WhtCtrl	用来决定亮区的对比度。数值为 256 时，亮区对比度不变。比 256 大时，值越大，亮区对比度越大；反之，比 256 小时，值越小，亮区对比度越小。 取值范围：[0x0, 0x200] 数据类型：CVI_U16
DciGainMax	用来控制对比度上限，数值越大，对比度可以越强。 取值范围：[0x0, 0x100] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDCIAttr

CVI\_ISP\_GetDCIAttr

## 22.3.2 ISP\_DCI\_AUTO\_ATTR\_S

**【说明】**

DCI 参数自动属性

**【定义】**

```
typedef struct ISP_DCI_AUTO_ATTR_S {  
    CVI_U16 ContrastGain[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BlcThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 WhtThr[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 BlcCtrl[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 WhtCtrl[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 DciGainMax[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_DCI_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
ContrastGain	用来控制 DCI 的强度，值越大，对比度越大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
BlcThr	用来决定暗区范围的门阈值。值越大，包含的暗区范围越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
WhtThr	用来决定亮区范围的门阈值。值越小，包含的亮区范围越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
BlcCtrl	用来决定暗区的对比度。数值为 256 时，暗区对比度不变。比 256 大时，值越大，暗区对比度越大；反之，比 256 小时，值越小，暗区对比度越小。 取值范围：[0x0, 0x200] 数据类型：CVI_U16
WhtCtrl	用来决定亮区的对比度。数值为 256 时，亮区对比度不变。比 256 大时，值越大，亮区对比度越大；反之，比 256 小时，值越小，亮区对比度越小。 取值范围：[0x0, 0x200] 数据类型：CVI_U16
DciGainMax	用来控制对比度上限，数值越大，对比度可以越强。 取值范围：[0x0, 0x100] 数据类型：CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDCIAttr

CVI\_ISP\_GetDCIAttr

### 22.3.3 ISP\_DCI\_ATTR\_S

**【说明】**

DCI 参数属性

**【定义】**

```
typedef struct _ISP_DCI_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_BOOL TuningMode;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_U8 Method;  
    CVI_U32 Speed;  
    CVI_U16 DciStrength;  
    CVI_U16 DciGamma;  
    CVI_U8 DciOffset;  
    CVI_U8 ToleranceY;  
    CVI_U8 Sensitivity;  
    ISP_DCI_MANUAL_ATTR_S stManual;  
    ISP_DCI_AUTO_ATTR_S stAuto;  
} ISP_DCI_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	DCI 模块使能。 0: 关闭。 1: 使能。
TuningMode	在同一画面同时显示原始图像和结果图像，辅助调节参数。 0: CAC 图像结果。 1: 在同一画面同时显示原始图像和结果图像。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
Method	用来选择 DCI algo 版本, 0: 自适应版本, 1: 手动调整版本 取值范围: [0x0, 0x1] 数据类型: CVI_U8
Speed	Smooth 强度, 值越高, 则变化越慢 取值范围: [0x0, 0x1f4] 数据类型: CVI_U16
DciStrength	保留每个 BIN 最低斜率, 值越大斜率越小 取值范围: [0x0, 0x100] 数据类型: CVI_U16
DciGamma	用来控制对比度, 数值越大, 对比越大, 但画面亮度会降低。 建议搭配 DciOffset 一起调试。 取值范围: [0x64, 0x320] 数据类型: CVI_U16
DciOffset	整体画面亮度的调整。数值越大, 整体亮度越大。建议搭配 DciGamma 一起调试。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
ToleranceY	到达目标亮度可容忍的亮度误差 取值范围: [0x0, 0xff] 数据类型: CVI_U8
Sensitivity	灵敏度: 灵敏度越高, dci 重运算的 threshold 越低 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDCIAttr

CVI\_ISP\_GetDCIAttr

# 23 LDCI

## 23.1 功能描述

Local Dynamic Contrast Improvement 的简称，藉由调整直方图来加强图像对比度，提升暗部细节。

## 23.2 API 参考

- `CVI_ISP_SetLDCIAttr`: 设置 LDCI 参数属性
- `CVI_ISP_GetLDCIAttr`: 获取 LDCI 参数属性

### 23.2.1 CVI\_ISP\_SetLDCIAttr

#### 【描述】

设置 LDCI 参数属性

#### 【语法】

```
CVI_S32 CVI_ISP_SetLDCIAttr(VI_PIPE ViPipe, const ISP_LDCI_ATTR_S *pstLDCIAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstLDCIAttr	DCI 参数属性	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_GetLDCIAttr](#)

## 23.2.2 CVI\_ISP\_GetLDCIAttr

**【描述】**

获取 LDCI 参数属性

**【语法】**

```
CVI_S32 CVI_ISP_GetLDCIAttr(VI_PIPE ViPipe, ISP_LDCI_ATTR_S *pstLDCIAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstLDCIAttr	DCI 参数属性	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

[CVI\\_ISP\\_SetLDCIAttr](#)

## 23.3 数据类型

- `ISP_LDCI_GAUSS_COEF_ATTR_S`: LDCI 高斯结构参数属性
- `ISP_LDCI_MANUAL_ATTR_S`: LDCI 参数手动属性
- `ISP_LDCI_AUTO_ATTR_S`: LDCI 参数自动属性
- `ISP_LDCI_ATTR_S`: LDCI 参数属性

### 23.3.1 `ISP_LDCI_GAUSS_COEF_ATTR_S`

#### 【说明】

LDCI 高斯结构参数属性

#### 【定义】

```
typedef struct _ISP_LDCI_GAUSS_COEF_ATTR_S {  
    CVI_U8 Wgt;  
    CVI_U8 Sigma;  
    CVI_U8 Mean;  
} ISP_LDCI_GAUSS_COEF_ATTR_S;
```

#### 【成员】

成员名称	描述
Wgt	根据亮度控制 LDCI 作用强度，决定 LDCI 结果与原图融合的权重曲线 取值范围：[0x0, 0x80] 数据类型：CVI_U8
Sigma	根据亮度控制 LDCI 作用强度，决定 LDCI 结果与原图融合的权重曲线 取值范围：[0x0, 0xff] 数据类型：CVI_U8
Mean	根据亮度控制 LDCI 作用强度，决定 LDCI 结果与原图融合的权重曲线 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无

#### 【相关数据类型及接口】

无



### 23.3.2 ISP\_LDCI\_MANUAL\_ATTR\_S

**【说明】**

LDCI 参数手动属性

**【定义】**

```
typedef struct _ISP_LDCI_MANUAL_ATTR_S {  
    CVI_U16 LdciStrength;  
    CVI_U16 LdciRange;  
    CVI_U16 TprCoef;  
    CVI_U8 EdgeCoring;  
    CVI_U8 LumaWgtMax;  
    CVI_U8 LumaWgtMin;  
    CVI_U8 VarMapMax;  
    CVI_U8 VarMapMin;  
    CVI_U8 UvGainMax;  
    CVI_U8 UvGainMin;  
    CVI_U8 BrightContrastHigh;  
    CVI_U8 BrightContrastLow;  
    CVI_U8 DarkContrastHigh;  
    CVI_U8 DarkContrastLow;  
    ISP_LDCI_GAUSS_COEF_ATTR_S LumaPosWgt;  
} ISP_LDCI_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
LdciStrength	控制 LDCI 增强效果参数。数值越大，局部对比度拉伸越强 取值范围：[0x0, 0x100] 数据类型：CVI_U16
LdciRange	针对图像的高频区，控制对比度增强的程度。数值越大，图像的高频区对比度越强 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
TprCoef	LD CI 曲线时间域上变化的平顺度。数值越小，时间域变化越平顺，反之，则变化越快 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
EdgeCoring	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMax	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMin	取值范围：[0x0, 0xff] 数据类型：CVI_U8
VarMapMax	取值范围：[0x0, 0xff] 数据类型：CVI_U8
VarMapMin	取值范围：[0x0, 0xff] 数据类型：CVI_U8
UvGainMax	取值范围：[0x0, 0x7f] 数据类型：CVI_U8
UvGainMin	取值范围：[0x0, 0x7f] 数据类型：CVI_U8
BrightContrastHigh	取值范围：[0x0, 0xff] 数据类型：CVI_U8
BrightContrastLow	取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkContrastHigh	取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkContrastLow	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaPosWgt	根据亮度控制 LDCI 作用强度，决定 LDCI 结果与原图融合的权重曲线

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetLDCIAttr

CVI\_ISP\_GetLDCIAttr

### 23.3.3 ISP\_LDCI\_AUTO\_ATTR\_S

**【说明】**

LDCI 参数自动属性

**【定义】**

```
typedef struct _ISP_LDCI_AUTO_ATTR_S {  
    CVI_U16 LdciStrength[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 LdciRange[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 TprCoef[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 LumaWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 VarMapMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 VarMapMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 UvGainMax[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 UvGainMin[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BrightContrastHigh[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 BrightContrastLow[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DarkContrastHigh[ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U8 DarkContrastLow[ISP_AUTO_ISO_STRENGTH_NUM];  
    ISP_LDCI_GAUSS_COEF_ATTR_S LumaPosWgt[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_LDCI_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
LdciStrength	控制 LDCI 增强效果参数。数值越大，局部对比度拉伸越强 取值范围：[0x0, 0x100] 数据类型：CVI_U16
LdciRange	针对图像的高频区，控制对比度增强的程度。数值越大，图像的高频区对比度越强 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
TprCoef	LD CI 曲线时间域上变化的平顺度。数值越小，时间域变化越平顺，反之，则变化越快 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
EdgeCoring	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMax	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaWgtMin	取值范围：[0x0, 0xff] 数据类型：CVI_U8
VarMapMax	取值范围：[0x0, 0xff] 数据类型：CVI_U8
VarMapMin	取值范围：[0x0, 0xff] 数据类型：CVI_U8
UvGainMax	取值范围：[0x0, 0x7f] 数据类型：CVI_U8
UvGainMin	取值范围：[0x0, 0x7f] 数据类型：CVI_U8
BrightContrastHigh	取值范围：[0x0, 0xff] 数据类型：CVI_U8
BrightContrastLow	取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkContrastHigh	取值范围：[0x0, 0xff] 数据类型：CVI_U8
DarkContrastLow	取值范围：[0x0, 0xff] 数据类型：CVI_U8
LumaPosWgt	根据亮度控制 LDCI 作用强度，决定 LDCI 结果与原图融合的权重曲线

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetLDCIAttr

CVI\_ISP\_GetLDCIAttr

### 23.3.4 ISP\_LDCI\_ATTR\_S

**【说明】**

DCI 参数属性

**【定义】**

```
typedef struct _ISP_LDCI_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_U8 GaussLPFSigma;  
    ISP_LDCI_MANUAL_ATTR_S stManual;  
    ISP_LDCI_AUTO_ATTR_S stAuto;  
} ISP_LDCI_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	LDCI 模块使能
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GaussLPFSigma	局部滤波程度，数值越小，局部对比度增强效果越局域化，反之，则越全局化 取值范围：[0x0, 0xff] 数据类型：CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetLDCIAttr

CVI\_ISP\_GetLDCIAttr

# 24 Dehaze

## 24.1 功能描述

藉由计算雾的浓度，从而自动调整对比度、饱和度以达到去雾的效果。

## 24.2 API 参考

- CVI\_ISP\_SetDehazeAttr: 设置去雾属性参数
- CVI\_ISP\_GetDehazeAttr: 获取去雾属性参数

### 24.2.1 CVI\_ISP\_SetDehazeAttr

#### 【描述】

设置去雾属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetDehazeAttr(VI_PIPE ViPipe, const ISP_DEHAZE_ATTR_S_
↪ *pstDehazeAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDehazeAttr	去雾属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

```
// Enable Dehaze and set strength as 100
VI_PIPE ViPipe = 0;
ISP_DEHAZE_ATTR_S attr;

CVI_ISP_GetDehazeAttr(ViPipe, &attr);
attr.Enable = CVI_TRUE;
attr.enOpType = OP_TYPE_AUTO;
attr.stAuto.Strength[0 /*ISO 100*/] = 100; // Strength=0-100
attr.stAuto.Strength[1 /*ISO 200*/] = 100; // Strength=0-100
...
attr.stAuto.Strength[15 /*ISO 3276800*/] = 100; // Strength=0-100
CVI_ISP_SetDehazeAttr(ViPipe, &attr);
```

**【相关主题】**

[CVI\\_ISP\\_GetDehazeAttr](#)

## 24.2.2 CVI\_ISP\_GetDehazeAttr

**【描述】**

获取去雾属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDehazeAttr(VI_PIPE ViPipe, ISP_DEHAZE_ATTR_S *pstDehazeAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDehazeAttr	去雾属性参数	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无

**【举例】**

无

**【相关主题】**

CVI\_ISP\_SetDehazeAttr

## 24.3 数据类型

- ISP\_DEHAZE\_MANUAL\_ATTR\_S: 去雾属性手动参数
- ISP\_DEHAZE\_AUTO\_ATTR\_S: 去雾属性自动参数
- ISP\_DEHAZE\_ATTR\_S: 去雾属性参数

### 24.3.1 ISP\_DEHAZE\_MANUAL\_ATTR\_S

**【说明】**

去雾属性手动参数

**【定义】**

```
typedef struct _ISP_DEHAZE_MANUAL_ATTR_S {  
    CVI_U8 Strength;  
} ISP_DEHAZE_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
Strength	用来控制 Dehaze 的强度。值越大，去雾强度越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDehazeAttr

CVI\_ISP\_GetDehazeAttr



## 24.3.2 ISP\_DEHAZE\_AUTO\_ATTR\_S

### 【说明】

去雾属性自动参数

### 【定义】

```
typedef struct _ISP_DEHAZE_AUTO_ATTR_S {
    CVI_U8 Strength[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DEHAZE_AUTO_ATTR_S;
```

### 【成员】

成员名称	描述
Strength	用来控制 Dehaze 的强度。值越大，去雾强度越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8

### 【注意事项】

无

### 【相关数据类型及接口】

CVI\_ISP\_SetDehazeAttr

CVI\_ISP\_GetDehazeAttr

## 24.3.3 ISP\_DEHAZE\_ATTR\_S

### 【说明】

去雾属性参数

### 【定义】

```
typedef struct _ISP_DEHAZE_ATTR_S {
    CVI_BOOL Enable; /*RW; Range:[0, 1]*/
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
    CVI_U16 CumulativeThr; /*RW; Range:[0x0, 0x3fff]*/
    CVI_U16 MinTransMapValue; /*RW; Range:[0x0, 0x1fff]*/
    CVI_BOOL DehazeLumaEnable; /*RW; Range:[0, 1]*/
    CVI_BOOL DehazeSkinEnable; /*RW; Range:[0, 1]*/
    CVI_U8 AirLightMixWgt; /*RW; Range:[0x0, 0x20]*/
    CVI_U8 DehazeWgt; /*RW; Range:[0x0, 0x20]*/
    CVI_U8 TransMapScale; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 AirlightDiffWgt; /*RW; Range:[0x0, 0x10]*/
    CVI_U16 AirLightMax; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 AirLightMin; /*RW; Range:[0x0, 0xffff]*/
    CVI_U8 SkinCb; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 SkinCr; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DehazeLumaCOEFFI; /*RW; Range:[0x0, 0x7d0]*/
```

(下页继续)

(续上页)

```
CVI_U16 DehazeSkinCOEFFI; /*RW; Range:[0x0, 0x7d0]*/  
CVI_U8 TransMapWgtWgt; /*RW; Range:[0x0, 0x80]*/  
CVI_U8 TransMapWgtSigma; /*RW; Range:[0x0, 0xff]*/  
ISP_DEHAZE_MANUAL_ATTR_S stManual;  
ISP_DEHAZE_AUTO_ATTR_S stAuto;  
} ISP_DEHAZE_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	Dehaze 功能使能 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好 取值范围: [0x0, 0xff] 数据类型: CVI_U8
CumulativeThr	计算雾浓度的统计阈值。默认值约为原始图像总像数个数的 0.05% 取值范围: [0x0, 0x3fff] 数据类型: CVI_U16
MinTransMapValue	透射系数允许之最小值 取值范围: [0x0, 0x1fff] 数据类型: CVI_U16
DehazeLumaEnable	“根据亮度调整去雾强度功能使能 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
DehazeSkinEnable	“根据肤色调整去雾强度功能使能 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
AirLightMixWgt	Airlight 混和权重 取值范围: [0x0, 0x20] 数据类型: CVI_U8
DehazeWgt	Dehaze 输出混和权重 取值范围: [0x0, 0x20] 数据类型: CVI_U8
TransMapScale	透射系数增益 取值范围: [0x0, 0xff] 数据类型: CVI_U8
AirlightDiffWgt	Airlight 取值范围: [0x0, 0x10] 数据类型: CVI_U8
AirLightMax	Airlight 可允许之最大值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
AirLightMin	Airlight 可允许之最小值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
SkinCb	自定义肤色在 Cb domain 上的坐标 取值范围: [0x0, 0xff] 数据类型: CVI_U8
SkinCr	自定义肤色在 Cr domain 上的坐标 取值范围: [0x0, 0xff] 数据类型: CVI_U8
DehazeLumaCOEFFI	根据亮度控制去雾强度曲线, 将亮度区分为 16 阶。值越大, 去雾强度越强 取值范围: [0x0, 0x7d0] 数据类型: CVI_U16
DehazeSkinCOEFFI	根据肤色控制去雾强度曲线, 将亮度区分为 16 阶。值越大, 去雾强度越强 取值范围: [0x0, 0x7d0] 数据类型: CVI_U16

**【注意事项】**

无

**【相关数据类型及接口】**

CVI\_ISP\_SetDehazeAttr

CVI\_ISP\_GetDehazeAttr

# 25 ColorTone

## 25.1 功能描述

在 WB 的基础之下，再额外调整图像颜色偏红或偏蓝等偏好。

## 25.2 API 参考

- CVI\_ISP\_SetColorToneAttr: 设置色调属性参数
- CVI\_ISP\_GetColorToneAttr: 获取色调属性参数

### 25.2.1 CVI\_ISP\_SetColorToneAttr

#### 【描述】

设置色调属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetColorToneAttr(VI_PIPE ViPipe, const ISP_COLOR_TONE_ATTR_S_
↪ *pstColorToneAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstColorToneAttr	色调属性参数	输入

#### 【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无

【举例】

无

【相关主题】

`CVI_ISP_GetColorToneAttr`

25.2.2 CVI\_ISP\_GetColorToneAttr

【描述】

获取色调属性参数

【语法】

```
CVI_S32 CVI_ISP_GetColorToneAttr(VI_PIPE ViPipe, ISP_COLOR_TONE_ATTR_S_
→*pstWBGAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstColorToneAttr	色调属性参数	输出

【返回值】

返回值	描述
0	成功
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无

【举例】

无

【相关主题】

`CVI_ISP_SetColorToneAttr`

## 25.3 数据类型

- `ISP_COLOR_TONE_ATTR_S`: 色调属性参数

### 25.3.1 `ISP_COLOR_TONE_ATTR_S`

#### 【说明】

色调属性参数

#### 【定义】

```
typedef struct _ISP_COLOR_TONE_ATTR_S {  
    CVI_U16 u16RedCastGain;  
    CVI_U16 u16GreenCastGain;  
    CVI_U16 u16BlueCastGain;  
} ISP_COLOR_TONE_ATTR_S;
```

#### 【成员】

成员名称	描述
u16RedCastGain	R 通道增益，8bit 小数精度。 取值范围：[0x0, 0x1000] 数据类型：CVI_U16
u16GreenCastGain	G 通道增益，8bit 小数精度。 取值范围：[0x0, 0x1000] 数据类型：CVI_U16
u16BlueCastGain	B 通道增益，8bit 小数精度。 取值范围：[0x0, 0x1000] 数据类型：CVI_U16

#### 【注意事项】

无

#### 【相关数据类型及接口】

`CVI_ISP_SetColorToneAttr`

`CVI_ISP_GetColorToneAttr`

# 26 Saturation

## 26.1 功能描述

调整颜色饱和度。

## 26.2 API 参考

- `CVI_ISP_SetSaturationAttr`：设置饱和度属性参数
- `CVI_ISP_GetSaturationAttr`：获取饱和度属性参数

### 26.2.1 CVI\_ISP\_SetSaturationAttr

#### 【描述】

设置饱和度属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetSaturationAttr(VI_PIPE ViPipe, const ISP_SATURATION_ATTR_S_
↪ *pstSaturationAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstSaturationAttr	饱和度属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】



- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetSaturationAttr](#)

## 26.2.2 CVI\_ISP\_GetSaturationAttr

**【描述】**

获取饱和度属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetSaturationAttr(VI_PIPE ViPipe, ISP_SATURATION_ATTR_S_
→*pstSaturationAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstSaturationAttr	饱和度属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetSaturationAttr](#)

## 26.3 数据类型

- `ISP_SATURATION_MANUAL_ATTR_S`：饱和度属性手动参数
- `ISP_SATURATION_AUTO_ATTR_S`：饱和度属性自动参数
- `ISP_SATURATION_ATTR_S`：饱和度属性参数

### 26.3.1 `ISP_SATURATION_MANUAL_ATTR_S`

#### 【说明】

饱和度属性手动参数

#### 【定义】

```
typedef struct _ISP_SATURATION_MANUAL_ATTR_S {  
    CVI_U8 Saturation;  
} ISP_SATURATION_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
Saturation	由三个数值组成的数组，定义输出的 UV 增益。 根据输入饱和度查找 UV 的增益，值越大，饱和度越高；反之，则越小。 取值范围：[0x0, 0xFF] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- `CVI_ISP_SetSaturationAttr`
- `CVI_ISP_GetSaturationAttr`

### 26.3.2 `ISP_SATURATION_AUTO_ATTR_S`

#### 【说明】

饱和度属性自动参数

#### 【定义】

```
typedef struct _ISP_SATURATION_AUTO_ATTR_S {  
    CVI_U8 Saturation[ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_SATURATION_AUTO_ATTR_S;
```

## 【成员】

成员名称	描述
Saturation	由三个数值组成的数组，定义输出的 UV 增益。 根据输入饱和度查找 UV 的增益，值越大，饱和度越高；反之，则越小。 取值范围：[0x0, 0xFF] 数据类型：CVI_U8

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetSaturationAttr
- CVI\_ISP\_GetSaturationAttr

### 26.3.3 ISP\_SATURATION\_ATTR\_S

## 【说明】

饱和度属性参数

## 【定义】

```
typedef struct _ISP_SATURATION_ATTR_S {  
    ISP_OP_TYPE_E enOpType;  
    ISP_SATURATION_MANUAL_ATTR_S stManual;  
    ISP_SATURATION_AUTO_ATTR_S stAuto;  
} ISP_SATURATION_ATTR_S;
```

## 【成员】

成员名称	描述
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
stAuto	自动模式参数属性
stManual	手动模式参数属性

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetSaturationAttr
- CVI\_ISP\_GetSaturationAttr

# 27 CAC

## 27.1 功能描述

色彩畸变校正, 因不同颜色的光折射率不同, 在高亮区与低亮区交界的物体周围容易形成紫边, 以此模块实现图像去紫边功能, 改善图像边缘的紫边现象。

## 27.2 API 参考

- `CVI_ISP_SetCacAttr` : 设置伽玛属性参数
- `CVI_ISP_GetCacAttr` : 获取伽玛属性参数

### 27.2.1 CVI\_ISP\_SetCacAttr

#### 【描述】

设置伽玛属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCacAttr(VI_PIPE ViPipe, const ISP_CAC_ATTR_S *pstCacAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCacAttr	伽玛属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetCacAttr](#)

## 27.2.2 CVI\_ISP\_GetCacAttr

**【描述】**

获取伽玛属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCacAttr(VI_PIPE ViPipe, ISP_CAC_ATTR_S *pstCacAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCacAttr	伽玛属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetCacAttr](#)

## 27.3 数据类型

- ISP\_CAC\_MANUAL\_ATTR\_S：伽玛属性手动参数
- ISP\_CAC\_AUTO\_ATTR\_S：伽玛属性自动参数
- ISP\_CAC\_ATTR\_S：伽玛属性参数

### 27.3.1 ISP\_CAC\_MANUAL\_ATTR\_S

#### 【说明】

伽玛属性手动参数

#### 【定义】

```
typedef struct ISP_CAC_MANUAL_ATTR_S {  
    CVI_U8 DePurpleStr; /*RW; Range:[0x0, 0xFF]*/  
    CVI_U8 EdgeGlobalGain; /*RW; Range:[0x0, 0xFF]*/  
    CVI_U8 EdgeCoring; /*RW; Range:[0x0, 0xFF]*/  
    CVI_U8 EdgeStrMin; /*RW; Range:[0x0, 0xFF]*/  
    CVI_U8 EdgeStrMax; /*RW; Range:[0x0, 0xFF]*/  
    CVI_U8 DePurpleCbStr; /*RW; Range:[0x0, 0x8]*/  
    CVI_U8 DePurpleCrStr; /*RW; Range:[0x0, 0x8]*/  
    CVI_U8 DePurpleStrMaxRatio; /*RW; Range:[0x0, 0x40]*/  
    CVI_U8 DePurpleStrMinRatio; /*RW; Range:[0x0, 0x40]*/  
} ISP_CAC_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
DePurpleStr	去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeGlobalGain	边缘侦测的强度总增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoring	边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeStrMin	边缘侦测的强度下限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeStrMax	边缘侦测的强度上限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleCbStr	B 通道紫边校正边强度。值越大，紫边现象越少 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DePurpleCrStr	R 通道紫边校正边强度。值越大，紫边现象越少 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DePurpleStrMaxRatio	去紫边强度可允许之最大增益 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurpleStrMinRatio	去紫边强度可允许之最小增益 取值范围：[0x0, 0x40] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCacAttr
- CVI\_ISP\_GetCacAttr

## 27.3.2 ISP\_CAC\_AUTO\_ATTR\_S

**【说明】**

伽玛属性自动参数

**【定义】**

```
typedef struct _ISP_CAC_AUTO_ATTR_S {
    CVI_U8 DePurpleStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeGlobalGain[ISP_AUTO_ISO_STRENGTH_NUM];
```

(下页继续)

(续上页)

```

CVI_U8 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 EdgeStrMin[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 EdgeStrMax[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 DePurpleCbStr[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 DePurpleCrStr[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 DePurpleStrMaxRatio[ISP_AUTO_ISO_STRENGTH_NUM];
CVI_U8 DePurpleStrMinRatio[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CAC_AUTO_ATTR_S;

```

**【成员】**

成员名称	描述
DePurpleStr	去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeGlobalGain	边缘侦测的强度总增益 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoring	边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeStrMin	边缘侦测的强度下限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeStrMax	边缘侦测的强度上限 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleCbStr	B 通道紫边校正边强度。值越大，紫边现象越少 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DePurpleCrStr	R 通道紫边校正边强度。值越大，紫边现象越少 取值范围：[0x0, 0x8] 数据类型：CVI_U8
DePurpleStrMaxRatio	去紫边强度可允许之最大增益 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurpleStrMinRatio	去紫边强度可允许之最小增益 取值范围：[0x0, 0x40] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCacAttr
- CVI\_ISP\_GetCacAttr



### 27.3.3 ISP\_CAC\_ATTR\_S

#### 【说明】

伽玛属性参数

#### 【定义】

```
typedef struct _ISP_CAC_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_U8 PurpleDetRange;  
    CVI_U8 PurpleCb;  
    CVI_U8 PurpleCr;  
    CVI_U8 PurpleCb2;  
    CVI_U8 PurpleCr2;  
    CVI_U8 PurpleCb3;  
    CVI_U8 PurpleCr3;  
    CVI_U8 GreenCb;  
    CVI_U8 GreenCr;  
    CVI_U8 TuningMode;  
    CVI_U8 EdgeGainIn[3];  
    CVI_U8 EdgeGainOut[3];  
    ISP_CAC_MANUAL_ATTR_S stManual;  
    ISP_CAC_AUTO_ATTR_S stAuto;  
} ISP_CAC_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	CAC 模块使能 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleDetRange	紫边检测的阈值。值越大，越多区域被判断为紫边 取值范围：[0x0, 0x80] 数据类型：CVI_U8
PurpleCb	紫色在 Cb domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleCr	紫色在 Cr domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleCb2	紫色 2 在 Cb domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleCr2	紫色 2 在 Cr domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleCb3	紫色 3 在 Cb domain 的坐标。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
PurpleCr3	紫色 3 在 Cr domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GreenCb	绿色在 Cb domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
GreenCr	绿色在 Cr domain 的坐标 取值范围：[0x0, 0xff] 数据类型：CVI_U8
TuningMode	输出元文件案，辅助调节参数。 0: CAC 图像结果。 1: 边缘检测图像结果。 2: 紫边检测图像结果 取值范围：[0x0, 0x2] 数据类型：CVI_U8
EdgeGainIn[3]	由四个数值组成的数组。定义边缘强度等级，值越大，边缘强度越强 取值范围：[0x0, 0x20] 数据类型：CVI_U8
EdgeGainOut[3]	由四个数值组成的数组。定义边缘强度增益，值越大，输出的边缘强度越强 取值范围：[0x0, 0x20] 数据类型：CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无

**【相关数据类型及接口】**

- CVI\_ISP\_SetCacAttr
- CVI\_ISP\_GetCacAttr

# 28 LCAC

## 28.1 功能描述

色彩畸变校正, 因不同颜色的光折射率不同, 在高亮区与低亮区交界的物体周围容易形成紫边, 以此模块实现图像去紫边功能, 改善图像边缘的紫边现象。

## 28.2 API 参考

- `CVI_ISP_SetLCACAttr` : 设置伽玛属性参数
- `CVI_ISP_GetLCACAttr` : 获取伽玛属性参数

### 28.2.1 CVI\_ISP\_SetLCACAttr

#### 【描述】

设置伽玛属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetLCACAttr(VI_PIPE ViPipe, const ISP_LCAC_ATTR_S *pstLCACAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstLCACAttr	伽玛属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetLCACAttr](#)

## 28.2.2 CVI\_ISP\_GetLCACAttr

**【描述】**

获取伽玛属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetLCACAttr(VI_PIPE ViPipe, ISP_LCAC_ATTR_S *pstLCACAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCacAttr	伽玛属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetLCACAttr](#)

## 28.3 数据类型

- `ISP_LCAC_GAUSS_COEF_ATTR_S`：伽玛高斯参数
- `ISP_LCAC_MANUAL_ATTR_S`：伽玛属性手动参数
- `ISP_LCAC_AUTO_ATTR_S`：伽玛属性自动参数
- `ISP_LCAC_ATTR_S`：伽玛属性参数

### 28.3.1 `ISP_LCAC_GAUSS_COEF_ATTR_S`

#### 【说明】

伽玛高斯参数

#### 【定义】

```
typedef struct _ISP_LCAC_GAUSS_COEF_ATTR_S {  
    CVI_U8 Wgt;  
    CVI_U8 Sigma;  
} ISP_LCAC_GAUSS_COEF_ATTR_S;
```

#### 【成员】

成员名称	描述
Wgt	根据边缘强度控制去紫边程度，决定去紫边结果与原图融合的权重曲线。Wgt 越大，融合原图的比例越大 取值范围：[0x0, 0x80] 数据类型：CVI_U8
Sigma	根据边缘强度控制去紫边程度，决定去紫边结果与原图融合的权重曲线。Sigma 越大，跟原图融合的像素数量越多 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

## 28.3.2 ISP\_LCAC\_MANUAL\_ATTR\_S

### 【说明】

伽玛属性手动参数

### 【定义】

```
typedef struct _ISP_LCAC_MANUAL_ATTR_S {
    CVI_U16 DePurpleCrGain;
    CVI_U16 DePurpleCbGain;
    CVI_U8 DePurepleCrWgt0;
    CVI_U8 DePurepleCbWgt0;
    CVI_U8 DePurepleCrWgt1;
    CVI_U8 DePurepleCbWgt1;
    CVI_U8 EdgeCoringBase;
    CVI_U8 EdgeCoringAdv;
} ISP_LCAC_MANUAL_ATTR_S;
```

### 【成员】

成员名称	描述
DePurpleCrGain	R 通道紫边校正增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurpleCbGain	B 通道紫边校正增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurepleCrWgt0	长帧 R 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCbWgt0	长帧 B 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCrWgt1	短帧 R 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCbWgt1	短帧 B 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
EdgeCoringBase	Base 边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoringAdv	Advance 边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8

### 【注意事项】

无。

### 【相关数据类型及接口】

- CVI\_ISP\_SetLCACAttr
- CVI\_ISP\_GetLCACAttr

### 28.3.3 ISP\_LCAC\_AUTO\_ATTR\_S

#### 【说明】

伽玛属性自动参数

#### 【定义】

```
typedef struct _ISP_LCAC_AUTO_ATTR_S {
    CVI_U16 DePurpleCrGain[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 DePurpleCbGain[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurepleCrWgt0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurepleCbWgt0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurepleCrWgt1[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurepleCbWgt1[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeCoringBase[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeCoringAdv[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_LCAC_AUTO_ATTR_S;
```

#### 【成员】

成员名称	描述
DePurpleCrGain	R 通道紫边校正增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurpleCbGain	B 通道紫边校正增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurepleCrWgt0	长帧 R 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCbWgt0	长帧 B 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCrWgt1	短帧 R 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurepleCbWgt1	短帧 B 通道校正强度混和权重 取值范围：[0x0, 0x40] 数据类型：CVI_U8
EdgeCoringBase	Base 边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoringAdv	Advance 边缘侦测的噪声抑制控制 取值范围：[0x0, 0xff] 数据类型：CVI_U8



**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetLCACAttr
- CVI\_ISP\_GetLCACAttr

### 28.3.4 ISP\_LCAC\_ATTR\_S

**【说明】**

伽玛属性参数

**【定义】**

```
typedef struct ISP_LCAC_ATTR_S {  
    CVI_BOOL Enable;  
    OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    CVI_U8 TuningMode;  
    CVI_U8 DePurpleCrStr0;  
    CVI_U8 DePurpleCbStr0;  
    CVI_U8 DePurpleCrStr1;  
    CVI_U8 DePurpleCbStr1;  
    CVI_U8 FilterTypeBase;  
    CVI_U8 EdgeGainBase0;  
    CVI_U8 EdgeGainBase1;  
    CVI_U8 EdgeStrWgtBase;  
    CVI_U8 DePurpleStrMaxBase;  
    CVI_U8 DePurpleStrMinBase;  
    CVI_U8 FilterScaleAdv;  
    CVI_U8 LumaWgt;  
    CVI_U8 FilterTypeAdv;  
    CVI_U8 EdgeGainAdv0;  
    CVI_U8 EdgeGainAdv1;  
    CVI_U8 EdgeStrWgtAdvG;  
    CVI_U8 DePurpleStrMaxAdv;  
    CVI_U8 DePurpleStrMinAdv;  
    ISP_LCAC_GAUSS_COEF_ATTR_S EdgeWgtBase;  
    ISP_LCAC_GAUSS_COEF_ATTR_S EdgeWgtAdv;  
    ISP_LCAC_MANUAL_ATTR_S stManual;  
    ISP_LCAC_AUTO_ATTR_S stAuto;  
} ISP_LCAC_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	LCAC 模块使能 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
TuningMode	边缘侦测的阈值。值越小，越多区域被判断为边缘 取值范围：[0x0, 0x6] 数据类型：CVI_U8
DePurpleCrStr0	长帧 R 通道校正强度 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurpleCbStr0	长帧 B 通道校正强度 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurpleCrStr1	短帧 R 通道校正强度 取值范围：[0x0, 0x40] 数据类型：CVI_U8
DePurpleCbStr1	短帧 B 通道校正强度 取值范围：[0x0, 0x40] 数据类型：CVI_U8
FilterTypeBase	滤波器选择。值越大，去紫边强度越强 取值范围：[0x0, 0x3] 数据类型：CVI_U8
EdgeGainBase0	长帧边缘侦测的强度增益 取值范围：[0x0, 0x1c] 数据类型：CVI_U8
EdgeGainBase1	短帧边缘侦测的强度增益 取值范围：[0x0, 0x23] 数据类型：CVI_U8
EdgeStrWgtBase	Base 通道边缘侦测权重 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleStrMaxBase	Base 去紫边强度的上限 取值范围：[0x0, 0x80] 数据类型：CVI_U8
DePurpleStrMinBase	Base 去紫边强度的下限 取值范围：[0x0, 0x80] 数据类型：CVI_U8
FilterScaleAdv	滤波器尺度参数 取值范围：[0x0, 0xf] 数据类型：CVI_U8
LumaWgt	亮度混和权重 取值范围：[0x0, 0x1ff] 数据类型：CVI_U16
FilterTypeAdv	滤波器选择。值越大，去紫边强度越强 取值范围：[0x0, 0x40] 数据类型：CVI_U8
EdgeGainAdv0	长帧边缘侦测的强度增益 取值范围：[0x0, 0x1c]

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetLCACAttr
- CVI\_ISP\_GetLCACAttr

# 29 RGBCAC

## 29.1 功能描述

实现图像去紫边功能，改善图像边缘的紫边现象。

## 29.2 API 参考

- `CVI_ISP_SetRGBCACAttr`：设置伽玛属性参数
- `CVI_ISP_GetRGBCACAttr`：获取伽玛属性参数

### 29.2.1 CVI\_ISP\_SetRGBCACAttr

#### 【描述】

设置伽玛属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetRGBCACAttr(VI_PIPE ViPipe, const ISP_RGBCAC_ATTR_S *  
    ↪pstRGBCACAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRGBCACAttr	伽玛属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_GetRGBCACAttr](#)

29.2.2 CVI\_ISP\_GetRGBCACAttr

【描述】

获取伽玛属性参数

【语法】

```
CVI_S32 CVI_ISP_GetRGBCACAttr(VI_PIPE ViPipe, ISP_RGBCAC_ATTR_S *pRGBCACAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstRGBCACAttr	伽玛属性参数	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_SetRGBCACAttr](#)

## 29.3 数据类型

- ISP\_RGBCAC\_MANUAL\_ATTR\_S：伽玛属性手动参数
- ISP\_RGBCAC\_AUTO\_ATTR\_S：伽玛属性自动参数
- ISP\_RGBCAC\_ATTR\_S：伽玛属性参数

### 29.3.1 ISP\_RGBCAC\_MANUAL\_ATTR\_S

#### 【说明】

伽玛属性手动参数

#### 【定义】

```
typedef struct ISP_RGBCAC_MANUAL_ATTR_S {
    CVI_U8 DePurpleStr0; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 DePurpleStr1; /*RW; Range:[0x0, 0xFF]*/
    CVI_U16 EdgeCoring; /*RW; Range:[0x0, 0xFFF]*/
    CVI_U8 DePurpleCrStr0; /*RW; Range:[0x0, 0x10]*/
    CVI_U8 DePurpleCbStr0; /*RW; Range:[0x0, 0x10]*/
    CVI_U8 DePurpleCrStr1; /*RW; Range:[0x0, 0x10]*/
    CVI_U8 DePurpleCbStr1; /*RW; Range:[0x0, 0x10]*/
} ISP_RGBCAC_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
DePurpleStr0	长帧去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleStr1	短帧去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoring	边缘侦测的噪声抑制控制 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurpleCrStr0	长帧 R 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCbStr0	长帧 B 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCrStr1	短帧 R 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCbStr1	短帧 B 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetRGBCACAttr
- CVI\_ISP\_GetRGBCACAttr

**29.3.2 ISP\_RGBCAC\_AUTO\_ATTR\_S****【说明】**

伽玛属性自动参数

**【定义】**

```
typedef struct ISP_RGBCAC_AUTO_ATTR_S {
    CVI_U8 DePurpleStr0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleStr1[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCrStr0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCbStr0[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCrStr1[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCbStr1[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_RGBCAC_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
DePurpleStr0	长帧去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleStr1	短帧去紫边强度。值越大，紫边现象越少 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeCoring	边缘侦测的噪声抑制控制 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
DePurpleCrStr0	长帧 R 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCbStr0	长帧 B 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCrStr1	短帧 R 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8
DePurpleCbStr1	短帧 B 通道校正边强度 取值范围：[0x0, 0x10] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetRGBCACAttr
- CVI\_ISP\_GetRGBCACAttr

### 29.3.3 ISP\_RGBCAC\_ATTR\_S

**【说明】**

伽玛属性参数

**【定义】**

```
typedef struct _ISP_RGBCAC_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U16 VarThr;  
    CVI_U8 PurpleDetRange0;  
    CVI_U8 PurpleDetRange1;  
    CVI_U8 DePurpleStrMax0;  
    CVI_U8 DePurpleStrMin0;  
    CVI_U8 DePurpleStrMax1;  
    CVI_U8 DePurpleStrMin1;  
    CVI_U16 EdgeGlobalGain;  
    CVI_U8 EdgeGainIn[3];  
    CVI_U8 EdgeGainOut[3];  
    CVI_U16 LumaScale;  
    CVI_U16 UserDefineLuma;  
    CVI_U8 LumaBlendWgt;  
    CVI_U8 LumaBlendWgt2;  
    CVI_U8 LumaBlendWgt3;  
    CVI_U8 PurpleCb;  
    CVI_U8 PurpleCr;  
    CVI_U8 PurpleCb2;  
    CVI_U8 PurpleCr2;  
    CVI_U8 PurpleCb3;  
    CVI_U8 PurpleCr3;  
    CVI_U8 GreenCb;  
    CVI_U8 GreenCr;  
    CVI_U8 TuningMode;  
    ISP_RGBCAC_MANUAL_ATTR_S stManual;  
    ISP_RGBCAC_AUTO_ATTR_S stAuto;  
} ISP_RGBCAC_ATTR_S;
```

**【成员】**



成员名称	描述
Enable	CAC 模块使能 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
VarThr	边缘侦测的阈值。值越小，越多区域被判断为边缘 取值范围：[0x0, 0x1ff] 数据类型：CVI_U16
PurpleDetRange0	长帧紫边侦测的阈值。值越大，越多区域被判断为紫边 取值范围：[0x0, 0x80] 数据类型：CVI_U8
PurpleDetRange1	短帧紫边侦测的阈值。值越大，越多区域被判断为紫边 取值范围：[0x0, 0x80] 数据类型：CVI_U8
DePurpleStrMax0	长帧去紫边强度可允许之最大值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleStrMin0	长帧去紫边强度可允许之最小值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleStrMax1	短帧去紫边强度可允许之最大值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
DePurpleStrMin1	短帧去紫边强度可允许之最小值 取值范围：[0x0, 0xff] 数据类型：CVI_U8
EdgeGlobalGain	边缘侦测的强度总增益 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
EdgeGainIn[3]	由四个数值组成的数组。定义边缘强度等级，值越大，边缘强度越强 取值范围：[0x0, 0x3] 数据类型：CVI_U8
EdgeGainOut[3]	由四个数值组成的数组。定义边缘强度增益，值越大，去紫边效果越强 取值范围：[0x0, 0x3] 数据类型：CVI_U8
LumaScale	亮度控制 取值范围：[0x0, 0x7ff] 数据类型：CVI_U16
UserDefineLuma	自定义亮度值 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
LumaBlendWgt	输入/输出
LumaBlendWgt2	亮度 2 的融合权重 取值范围：[0x0, 0x20] 数据类型：CVI_U8
LumaBlendWgt3	亮度 3 的融合权重 取值范围：[0x0, 0x20] 数据类型：CVI_U8
PurpleCb	紫色在 Cb domain 的坐标 取值范围：[0x0, 0xffff]

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetRGBCACAttr
- CVI\_ISP\_GetRGBCACAttr

# 30 FSWDR

## 30.1 功能描述

宽动态范围合成相关参数。

## 30.2 API 参考

- `CVI_ISP_SetFSWDRAttr`：设置帧合成属性参数
- `CVI_ISP_GetFSWDRAttr`：获取帧合成属性参数
- `CVI_ISP_SetWDRExposureAttr`：设置 WDR 曝光属性参数
- `CVI_ISP_GetWDRExposureAttr`：获取 WDR 曝光属性参数

### 30.2.1 CVI\_ISP\_SetFSWDRAttr

#### 【描述】

设置帧合成属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetFSWDRAttr(VI_PIPE ViPipe, const ISP_FSWDR_ATTR_S_
↪ *pstFSWDRAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstFSWDRAttr	帧合成属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

- `cv180x` 不支持此功能
- 开启 `wdr` 模式时, 建议将 `Gamma table` 设为 `sRGB`

**【举例】**

无。

**【相关主题】**

- `CVI_ISP_GetFSWDRAttr`

## 30.2.2 CVI\_ISP\_GetFSWDRAttr

**【描述】**

获取帧合成属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetFSWDRAttr(VI_PIPE ViPipe, ISP_FSWDR_ATTR_S *pstFSWDRAttr);
```

**【参数】**

参数名称	描述	输入/输出
<code>ViPipe</code>	<code>VI_PIPE</code> 号	输入
<code>pstFSWDRAttr</code>	帧合成属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

- `cv180x` 不支持此功能

**【举例】**

- 开启 `wdr` 模式时, 建议将 `Gamma table` 设为 `sRGB`

**【相关主题】**

- [CVI\\_ISP\\_SetFSWDRAttr](#)

### 30.2.3 CVI\_ISP\_SetWDRExposureAttr

#### 【描述】

设置 WDR 曝光属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetWDRExposureAttr(VI_PIPE ViPipe, const ISP_WDR_EXPOSURE_
→ATTR_S *pstWDRExpAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstWDRExposureAttr	WDR 曝光属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

#### 【注意】

- `cv180x` 不支持此功能

#### 【举例】

- 开启 `wdr` 模式时, 建议将 `Gamma table` 设为 `sRGB`

#### 【相关主题】

- [CVI\\_ISP\\_GetWDRExposureAttr](#)

### 30.2.4 CVI\_ISP\_GetWDRExposureAttr

**【描述】**

获取 WDR 曝光属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetWDRExposureAttr(VI_PIPE ViPipe, ISP_WDR_EXPOSURE_ATTR_S_
→*pstWDRExposureAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstWDRExposureAttr	WDR 曝光属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

- cv180x 不支持此功能
- 开启 wdr 模式时, 建议将 Gamma table 设为 sRGB

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_SetWDRExposureAttr

## 30.3 数据类型

- ISP\_FSWDR\_MANUAL\_ATTR\_S : 帧合成属性手动参数
- ISP\_FSWDR\_AUTO\_ATTR\_S : 帧合成属性自动参数
- ISP\_FSWDR\_ATTR\_S : 帧合成属性参数
- ISP\_WDR\_EXPOSURE\_ATTR\_S : WDR 曝光属性参数

### 30.3.1 ISP\_FSWDR\_MANUAL\_ATTR\_S

**【说明】**

帧合成属性手动参数

**【定义】**

```
typedef struct _ISP_FSWDR_MANUAL_ATTR_S {  
    CVI_U16 WDRCombineLongThr; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRCombineShortThr; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRCombineMaxWeight; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRCombineMinWeight; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 WDRMtIn[4]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U16 WDRMtOut[4]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRLongWgt; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 WDRCombineSNRAwareToleranceLevel; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 MergeModeAlpha; /*RW; Range:[0x0, 0xff]*/  
    CVI_U16 WDRMotionCombineLongThr; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRMotionCombineShortThr; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRMotionCombineMinWeight; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRMotionCombineMaxWeight; /*RW; Range:[0x0, 0x100]*/  
} ISP_FSWDR_MANUAL_ATTR_S;
```

**【成员】**

成员名称	描述
WDRCombineLongThr	长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 影像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRCombineShortThr	短曝光临界值，超过该临界值的图像数据将只选择短曝光数据合成 WDR 影像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRCombineMinWeight	长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。（此值必须小于等于 WDRCombineMaxWeight） 取值范围：[0x0, 0x100] 数据类型：CVI_U8
WDRCombineMaxWeight	长短曝图像数据融合最高权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。（此值必须大于等于 WDRCombineMinWeight） 取值范围：[0x0, 0x100] 数据类型：CVI_U8
WDRMotionCombine-LongThr	运动侦测信息，长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 图像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRMotionCombineShort-Thr	运动侦测信息，短曝光临界值，超过该临界值的图像数据将只选择短曝光数据合成 WDR 图像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRMotionCombineMinWeight	运动侦测信息，长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
WDRMotionCombineMaxWeight	运动侦测信息，长短曝图像数据融合最高权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
WDRMtIn[4]	由四个数值组成的数组。定义输入物体运动量，值越大，运动量越大。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
WDRMtOut[4]	由四个数值组成的数组。定义输入物体运动量对应的运动增益，值越大，越倾向输出自定义的 WDR 融合结果，与 WDRType 配合调试。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
WDRLongWgt	当选择 WDRType 设定为 2 时，设定长曝的融合权重。MotionCompEnable 使能时才生效。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
MergeModeAlpha	MergeMode 设为为 1 时的移动信息融合比例。值越大，融合时短帧移动信息占的比重越多。反之则长帧移动信息占的比重越多。 取值范围：[0x0, 0xffff] 数据类型：CVI_U8
WDRCombineSNRAware-	短帧噪声 SNR 自适应之噪声容忍强度



**【注意事项】**

- 以上参数 cv180x 不支持

**【相关数据类型及接口】**

- CVI\_ISP\_SetFSWDRAttr
- CVI\_ISP\_GetFSWDRAttr

### 30.3.2 ISP\_FSWDR\_AUTO\_ATTR\_S

**【说明】**

帧合成属性自动参数

**【定义】**

```
typedef struct cviISP_FSWDRAttr_AUTO_ATTR_S {  
    CVI_U16 WDRCombineLongThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRCombineShortThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRCombineMaxWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRCombineMinWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 WDRMtIn[4][ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U16 WDRMtOut[4][ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRLongWgt[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 WDRCombineSNRAwareToleranceLevel[ISP_AUTO_LV_NUM];  
    CVI_U8 MergeModeAlpha[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/  
    CVI_U16 WDRMotionCombineLongThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRMotionCombineShortThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xffff]*/  
    CVI_U16 WDRMotionCombineMinWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
    CVI_U16 WDRMotionCombineMaxWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/  
}ISP_FSWDR_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
WDRCombineLongThr	长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 影像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRCombineMinWeight	长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。（此值必须小于等于 WDRCombineMaxWeight） 取值范围：[0x0, 0xff] 数据类型：CVI_U8
WDRMotionCombine-LongThr	运动侦测信息，长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 图像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRMotionCombineShort-Thr	运动侦测信息，短曝光临界值，超过该临界值的图像数据将只选择短曝光数据合成 WDR 图像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRMotionCombineMinWeight	运动侦测信息，长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
WDRMotionCombineMaxWeight	运动侦测信息，长短曝图像数据融合最高权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
MergeModeAlpha	MergeMode 设为为 1 时的移动信息融合比例。值越大，融合时短帧移动信息占的比重越多。反之则长帧移动信息占的比重越多。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
WDRCombineSNRAware-ToleranceLevel	短帧噪声 SNR 自适应之噪声容忍强度 取值范围：[0x0, 0xBB8] 数据类型：CVI_U16
WDRCombineLongThr	长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 影像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRCombineShortThr	短曝光临界值，超过该临界值的图像数据将只选择短曝光数据合成 WDR 影像。 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
WDRCombineMinWeight	长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。（此值必须小于等于 WDRCombineMaxWeight） 取值范围：[0x0, 0x100] 数据类型：CVI_U8
WDRCombineMaxWeight	长短曝图像数据融合最高权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多。（此值必须大于等于 WDRCombineMinWeight） 取值范围：[0x0, 0x100] 数据类型：CVI_U8
WDRMotionCombine-LongThr	运动侦测信息，长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 图像。

**【注意事项】**

- 以上参数 cv180x 不支持

**【相关数据类型及接口】**

- CVI\_ISP\_SetFSWDRAttr
- CVI\_ISP\_GetFSWDRAttr

### 30.3.3 ISP\_FSWDR\_ATTR\_S

**【说明】**

WDR 曝光属性参数

**【定义】**

```
typedef struct _ISP_FSWDR_ATTR_S {
    CVI_BOOL Enable; /*RW; Range:[0, 1]*/
    ISP_OP_TYPE_E enOpType; /*RW; Range:[0, 1]*/
    CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
    CVI_BOOL MotionCompEnable; /*RW; Range:[0, 1]*/
    CVI_U8 TuningMode; /*RW; Range:[0x0, 0x9]*/
    CVI_BOOL WDRDCMode;; /*RW; Range:[0, 1]*/
    CVI_BOOL WDRLumaMode; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 WDRType; /*RW; Range:[0x0, 0x2]*/
    CVI_BOOL WDRCombineSNRAwareEn; /*RW; Range:[0, 1]*/
    CVI_U16 WDRCombineSNRAwareLowThr; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 WDRCombineSNRAwareHighThr; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 WDRCombineSNRAwareSmoothLevel; /*RW; Range:[0x0, 0xbb8]*/
    CVI_BOOL LocalToneRefinedDCMode; /*RW; Range:[0, 1]*/
    CVI_BOOL LocalToneRefinedLumaMode; /*RW; Range:[0, 1]*/
    CVI_U16 DarkToneRefinedThrL; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DarkToneRefinedThrH; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DarkToneRefinedMaxWeight; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DarkToneRefinedMinWeight; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 BrightToneRefinedThrL; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 BrightToneRefinedThrH; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 BrightToneRefinedMaxWeight; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 BrightToneRefinedMinWeight; /*RW; Range:[0x0, 0x100]*/
    CVI_U8 WDRMotionFusionMode; /*RW; Range:[0x0, 0x3]*/
    CVI_BOOL MtMode; /*RW; Range:[0, 1]*/

    ISP_FSWDR_MANUAL_ATTR_S stManual;
    ISP_FSWDR_AUTO_ATTR_S stAuto;
} ISP_FSWDR_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	WDR 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
MotionCompEnable	WDR 运动侦测使能开关。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
WDRDCMode	长短曝融合模式 0: 一般模式, 透过 WDRLumaMode 调整 1: DC 模式 取值范围: [0, 1] 数据类型: CVI_BOOL
WDRLumaMode	亮度计算模式 0: 取 R/G/B 最大值。 1: 取 Luma 值。 取值范围: [0, 1] 数据类型: CVI_BOOL
WDRType	自定义 WDR 融合模式。 0: 输出乘上曝光比的短曝值。 1: 输出长曝值。 2: 输出长短曝融合后的数值。长曝的权重为 WDRLongWgt。 MotionCompEnable 使能时才生效。 取值范围: [0, 2] 数据类型: CVI_U8
WDRDitherEnable	针对长曝临界值做 dither 功能使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_U8
WDRDitherBit	设定产生几 bit 的 dither。 取值范围: [0, 0xf] 数据类型: CVI_U8
WDRDitherRange	设定在长短曝临界值加上 dither 的范围。 取值范围: [0, 0xffff] 数据类型: CVI_U16
TuningMode	调整模式, 输出可视化辅助信息, 帮助用户调试 0: 不输出可视化辅助信息 1: 以 SDR 形式输出短帧信息。 2: 以 HDR 形式输出短帧信息。 3: 以 SDR 形式输出长帧信息。 取值范围: [0x0, 0x3] 数据类型: CVI_U8
WDRCombineSNRAwareEn	具 SNR 感知的宽动态融合模式使能。能根据短帧噪声自适应 取决长短帧融合比例。 0: 关闭。 1: 使能。 取值范围: [0, 1]

**【注意事项】**

- 以上参数 cv180x 不支持

**【相关数据类型及接口】**

- CVI\_ISP\_SetFSWDRAttr
- CVI\_ISP\_GetFSWDRAttr

### 30.3.4 ISP\_WDR\_EXPOSURE\_ATTR\_S

**【说明】**

WDR 曝光属性参数

**【定义】**

```
typedef struct ISP_WDR_EXPOSURE_ATTR_S {  
    ISP_OP_TYPE_E enExpRatioType;  
    CVI_U32 au32ExpRatio[WDR_EXP_RATIO_NUM];  
    CVI_U32 u32ExpRatioMax;  
    CVI_U32 u32ExpRatioMin;  
    CVI_U16 u16Tolerance;  
    CVI_U16 u16Speed;  
    CVI_U16 u16RatioBias;  
    CVI_U8 u8SECompensation;  
    CVI_U16 u16SEHisThr;  
    CVI_U16 u16SEHisCntRatio1;  
    CVI_U16 u16SEHisCntRatio2;  
    CVI_U32 u16SEHis255CntThr1;  
    CVI_U32 u16SEHis255CntThr2;  
    CVI_U8 au8LEAdjustTargetMin[LV_TOTAL_NUM];  
    CVI_U8 au8LEAdjustTargetMax[LV_TOTAL_NUM];  
    CVI_U8 au8SEAdjustTargetMin[LV_TOTAL_NUM];  
    CVI_U8 au8SEAdjustTargetMax[LV_TOTAL_NUM];  
    CVI_U8 u8AdjustTargetDetectFrmNum;  
    CVI_U32 u32DiffPixelNum;  
    CVI_U16 u16LELowBinThr;  
    CVI_U16 u16LEHighBinThr;  
    CVI_U16 u16SELowBinThr;  
    CVI_U16 u16SEHighBinThr;  
    CVI_U8 au8FrameAvgLumaMin[LV_TOTAL_NUM];  
    CVI_U8 au8FrameAvgLumaMax[LV_TOTAL_NUM];  
} ISP_WDR_EXPOSURE_ATTR_S;
```

**【成员】**

成员名称	描述
enExpRatioType	<p>仅在多帧合成 WDR 模式下有效。</p> <p>OP_TYPE_AUTO: 根据场景自动计算长短帧曝光比;</p> <p>OP_TYPE_MANUAL: 手动配置长短帧曝光比。</p>
au32ExpRatio[WDR_EXP_RATIO]	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, au32ExpRatio 无效。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, au32ExpRatio 为可擦写, 表示多帧合成 WDR 相邻 2 帧曝光比期望值。</p> <p>取值范围: [0x40, 0x4000]</p> <p>数据类型: CVI_U32</p>
u32ExpRatioMax	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMax 表示最长帧与最短帧曝光时间比值的最大值。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMax 无效。</p> <p>6bit 小数精度, 0x40 表示曝光比为 1 倍。</p> <p>取值范围: [0x40, 0x4000]</p> <p>数据类型: CVI_U32</p>
u32ExpRatioMin	<p>仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMin 表示长帧曝光时间与短帧曝光时间比值的最小值。</p> <p>当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMin 无效。</p> <p>格式为无符号 6.6bit 定点, 0x40 表示长帧曝光时间与短帧曝光时间的比值为 1 倍。</p> <p>默认值为 0x40。</p> <p>取值范围: [0x40, 0x4000]</p> <p>数据类型: CVI_U32</p>
u16Tolerance	<p>曝光比容忍值, 仅在两帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 表示场景动态范围变化在一定范围内时, 曝光比保持不变。</p> <p>默认值为 0xC。</p> <p>取值范围: [0x0, 0xFF]</p> <p>数据类型: CVI_U16</p>
u16Speed	<p>自动曝光比调节速度, 仅在两帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 自动曝光比调节速度越快。</p> <p>默认值为 0x20。</p> <p>取值范围: [0x0, 0xFF]</p> <p>数据类型: CVI_U16</p>
u16RatioBias	<p>曝光比偏差值, 仅在多帧合成 WDR 模式下有效。</p> <p>当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 自动曝光比越大。</p> <p>默认值为 0x400, 表示不对自动曝光比算法的计算结果进行调整。</p> <p>经过该值调整的曝光比会受到曝光比最大/最小值的限制。</p> <p>取值范围: [0x0, 0xFFFF]</p> <p>数据类型: CVI_U16</p>
u8SECompensation	<p>调整短帧画面的目标亮度值</p> <p>取值范围: [0x0, 0xFF]</p>

**【注意事项】**

- 以上参数 cv180x 不支持

**【相关数据类型及接口】**

- CVI\_ISP\_SetWDRExposureAttr
- CVI\_ISP\_GetWDRExposureAttr

# 31 DRC

## 31.1 功能描述

整幅图像的动态范围，使之能在显示设备上的显示效果与人眼视觉感受一致。

## 31.2 API 参考

- `CVI_ISP_SetDRCAttr`：设置 DRC 属性参数
- `CVI_ISP_GetDRCAttr`：获取 DRC 属性参数

### 31.2.1 CVI\_ISP\_SetDRCAttr

#### 【描述】

设置 DRC 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetDRCAttr(VI_PIPE ViPipe, const ISP_DRC_ATTR_S *pstDRCAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDRCAttr	DRC 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`



- 库文件: libisp.so

**【注意】**

开启 wdr 模式时, 建议将 Gamma table 设为 sRGB

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetDRCAttr](#)

## 31.2.2 CVI\_ISP\_GetDRCAttr

**【描述】**

获取 DRC 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDRCAttr(VI_PIPE ViPipe, ISP_DRC_ATTR_S *pstDRCAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDRCAttr	DRC 属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

开启 wdr 模式时, 建议将 Gamma table 设为 sRGB

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetDRCAttr](#)

## 31.3 数据类型

- ISP\_DRC\_MANUAL\_ATTR\_S : DRC 属性手动参数
- ISP\_DRC\_AUTO\_ATTR\_S : DRC 属性自动参数
- ISP\_DRC\_ATTR\_S : DRC 属性参数

### 31.3.1 ISP\_DRC\_MANUAL\_ATTR\_S

#### 【说明】

DRC 属性手动参数

#### 【定义】

```
typedef struct ISP_DRC_MANUAL_ATTR_S {  
    CVI_U32 TargetYScale; /*RW; Range:[0x0, 0x800]*/  
    CVI_U16 HdrStrength; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 DEAdaptPercentile; /*RW; Range:[0x0, 0x19]*/  
    CVI_U8 DEAdaptTargetGain; /*RW; Range:[0x1, 0x60]*/  
    CVI_U8 DEAdaptGainUB; /*RW; Range:[0x1, 0xff]*/  
    CVI_U8 DEAdaptGainLB; /*RW; Range:[0x1, 0xff]*/  
    CVI_U8 BritInflectPtLuma; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 BritContrastLow; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 BritContrastHigh; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 SdrTargetY; /*RW; Range:[0x0, 0xff]*/  
    CVI_U8 SdrTargetYGain; /*RW; Range:[0x20, 0x80]*/  
    CVI_U16 SdrGlobalToneStr; /*RW; Range:[0x0, 0x100]*/  
    CVI_U8 SdrDEAdaptPercentile; /*RW; Range:[0x0, 0x19]*/  
    CVI_U8 SdrDEAdaptTargetGain; /*RW; Range:[0x1, 0x40]*/  
    CVI_U8 SdrDEAdaptGainLB; /*RW; Range:[0x1, 0xff]*/  
    CVI_U8 SdrDEAdaptGainUB; /*RW; Range:[0x1, 0xff]*/  
    CVI_U8 SdrBritInflectPtLuma; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 SdrBritContrastLow; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 SdrBritContrastHigh; /*RW; Range:[0x0, 0x64]*/  
    CVI_U8 TotalGain; /*RW; Range:[0x0, 0xff]*/  
} ISP_DRC_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
LETargetYScale	暗部色调映像增强参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围： [0x0, 0x800] 数据类型： CVI_U32

下页继续

表 31.1 – 续上页

成员名称	描述
TargetYScale	参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
HdrStrength	控制 HDR 增强效果参数，数值越大整体通透度拉伸越强，反之则拉伸越弱。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DarkOffset	控制暗区提亮，数值越大暗区提亮越多，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
ContrastDarkMinThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMaxThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMinWeight	局部对比增强：用来控制暗区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastDarkMaxWeight	局部对比增强：用来控制暗区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMinThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMaxThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMinWeight	局部对比增强：用来控制亮区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMaxWeight	局部对比增强：用来控制亮区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

下页继续

表 31.1 – 续上页

成员名称	描述
ContrastGain	全局对比控制参数，128 为 1 倍，值越小于 128 对比度降低越多，值越大 128 对比度提升越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DarkToneRange	控制暗区提亮范围值，数值越大暗区提亮越大，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DEAdaptPercentile	定义 Dark 的百分位数，决定不加强的区域。 取值范围：[0x0, 0x19] 数据类型：CVI_U8
DEAdaptTargetGain	Dark Tone 自适应目标的强化，值越大拉的越亮，32 为一倍，40 为 1.25x。 取值范围：[0x1, 0x40] 数据类型：CVI_U8
DEAdaptGainLB	Dark Tone 自适应强化的下界，值越大越不压黑，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
DEAdaptGainUB	Dark Tone 自适应强化的上界，值越大拉的越亮，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
SdrDEAdaptPercentile	定义 Dark 的百分位数，决定不加强的区域。 取值范围：[0x0, 0x19] 数据类型：CVI_U8
SdrDEAdaptTargetGain	Dark Tone 自适应目标的强化，值越大拉的越亮，32 为一倍，40 为 1.25x。 取值范围：[0x1, 0x40] 数据类型：CVI_U8
SdrDEAdaptGainLB	Dark Tone 自适应强化的下界，值越大越不压黑，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
SdrDEAdaptGainUB	Dark Tone 自适应强化的上界，值越大拉的越亮，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
BritInfectPtLuma	Bright Tone 在长短曝交界区的亮度，值越大，亮度越高。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
BritContrastLow	Bright Tone 亮区中的暗处压暗程度，值越大，压暗越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
BritContrastHigh	Bright Tone 亮区中亮处拉亮程度，值越大，拉升越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8

下页继续

表 31.1 – 续上页

成员名称	描述
SdrTargetY	全局性以 Global Tone 将画面拉亮，值越大，亮度拉高越多。 取值范围：[0x0, 0xFF] 数据类型：CVI_U8
SdrTargetYGain	全局性以 Global Tone 将画面拉亮，目标亮度采用当前平均亮度的倍数， $1x=32, 2x=64$ 。 取值范围：[0x20, 0x80] 数据类型：CVI_U8
SdrGlobalToneStr	Global Tone 的强度，值越大，Global Tone 越强，反之则往 Linear Tone 靠近。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
SdrBritInfectPtLuma	Bright Tone 在长短曝交界区的亮度，值越大，亮度越高。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
SdrBritContrastLow	Bright Tone 亮区中的暗处压暗程度，值越大，压暗越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
SdrBritContrastHigh	Bright Tone 亮区中亮处拉亮程度，值越大，拉升越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
LETargetYScale	暗部色调映像增强参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
TargetYScale	参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
HdrStrength	控制 HDR 增强效果参数，数值越大整体通透度拉伸越强，反之则拉伸越弱。 取值范围：[0x0, 0xff] 数据类型：CVI_U16
DarkOffset	控制暗区提亮，数值越大暗区提亮越多，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMinThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMaxThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16

下页继续

表 31.1 – 续上页

成员名称	描述
ContrastDarkMinWeight	局部对比增强：用来控制暗区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastDarkMaxWeight	局部对比增强：用来控制暗区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMinThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMaxThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMinWeight	局部对比增强：用来控制亮区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMaxWeight	局部对比增强：用来控制亮区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
TotalGain	整体的细节强度，1x = 32 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDRCAttr
- CVI\_ISP\_GetDRCAttr

**31.3.2 ISP\_DRC\_AUTO\_ATTR\_S****【说明】**

DRC 属性自动参数

**【定义】**

```
typedef struct _ISP_DRC_AUTO_ATTR_S {
    CVI_U32 TargetYScale[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x800]*/
    CVI_U16 HdrStrength[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
}
```

(下页继续)

(续上页)

```

CVI_U8 DEAdaptPercentile[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x19]*/
CVI_U8 DEAdaptTargetGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0x60]*/
CVI_U8 DEAdaptGainUB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
CVI_U8 DEAdaptGainLB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
CVI_U8 BritInflectPtLuma[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 BritContrastLow[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 BritContrastHigh[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 SdrTargetY[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/
CVI_U8 SdrTargetYGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x20, 0x80]*/
CVI_U16 SdrGlobalToneStr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
CVI_U8 SdrDEAdaptPercentile[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x19]*/
CVI_U8 SdrDEAdaptTargetGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0x40]*/
CVI_U8 SdrDEAdaptGainLB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
CVI_U8 SdrDEAdaptGainUB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
CVI_U8 SdrBritInflectPtLuma[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 SdrBritContrastLow[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 SdrBritContrastHigh[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
CVI_U8 TotalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
} ISP_DRC_AUTO_ATTR_S;

```

## 【成员】

成员名称	描述
LETARGETYScale	暗部色调映像增强参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
TargetYScale	参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
HdrStrength	控制 HDR 增强效果参数，数值越大整体通透度拉伸越强，反之则拉伸越弱。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DarkOffset	控制暗区提亮，数值越大暗区提亮越多，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
ContrastDarkMinThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMaxThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16

下页继续



表 31.2 – 续上页

成员名称	描述
ContrastDarkMinWeight	局部对比增强：用来控制暗区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastDarkMaxWeight	局部对比增强：用来控制暗区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMinThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMaxThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMinWeight	局部对比增强：用来控制亮区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMaxWeight	局部对比增强：用来控制亮区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastGain	全局对比控制参数，128 为 1 倍，值越小于 128 对比度降低越多，值越大 128 对比度提升越多。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DarkToneRange	控制暗区提亮范围值，数值越大暗区提亮越大，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
DEAdaptPercentile	定义 Dark 的百分位数，决定不加强的区域。 取值范围：[0x0, 0x19] 数据类型：CVI_U8
DEAdaptTargetGain	Dark Tone 自适应目标的强化，值越大拉的越亮，32 为一倍，40 为 1.25x。 取值范围：[0x1, 0x40] 数据类型：CVI_U8
DEAdaptGainLB	Dark Tone 自适应强化的下界，值越大越不压黑，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
DEAdaptGainUB	Dark Tone 自适应强化的上界，值越大拉的越亮，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8

下页继续



表 31.2 – 续上页

成员名称	描述
SdrDEAdaptPercentile	定义 Dark 的百分位数，决定不加强的区域。 取值范围：[0x0, 0x19] 数据类型：CVI_U8
SdrDEAdaptTargetGain	Dark Tone 自适应目标的强化，值越大拉的越亮，32 为一倍，40 为 1.25x。 取值范围：[0x1, 0x40] 数据类型：CVI_U8
SdrDEAdaptGainLB	Dark Tone 自适应强化的下界，值越大越不压黑，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
SdrDEAdaptGainUB	Dark Tone 自适应强化的上界，值越大拉的越亮，32 为一倍，96 为 3x。 取值范围：[0x1, 0xFF] 数据类型：CVI_U8
BritInfectPtLuma	Bright Tone 在长短曝交界区的亮度，值越大，亮度越高。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
BritContrastLow	Bright Tone 亮区中的暗处压暗程度，值越大，压暗越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
BritContrastHigh	Bright Tone 亮区中亮处拉亮程度，值越大，拉升越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
SdrTargetY	全局性以 Global Tone 将画面拉亮，值越大，亮度拉高越多。 取值范围：[0x0, 0xFF] 数据类型：CVI_U8
SdrTargetYGain	全局性以 Global Tone 将画面拉亮，目标亮度采用当前平均亮度的倍数，1x=32, 2x = 64。 取值范围：[0x20, 0x80] 数据类型：CVI_U8
SdrGlobalToneStr	Global Tone 的强度，值越大，Global Tone 越强，反之则往 Linear Tone 靠近。 取值范围：[0x0, 0x100] 数据类型：CVI_U16
SdrBritInfectPtLuma	Bright Tone 在长短曝交界区的亮度，值越大，亮度越高。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
SdrBritContrastLow	Bright Tone 亮区中的暗处压暗程度，值越大，压暗越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
SdrBritContrastHigh	Bright Tone 亮区中亮处拉亮程度，值越大，拉升越多。 取值范围：[0x0, 0x64] 数据类型：CVI_U8

下页继续

表 31.2 – 续上页

成员名称	描述
LETargetYScale	暗部色调映像增强参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
TargetYScale	参考目标亮度值增益，drc 根据该值与当前宽动态亮度分布平均值产生参考目标亮度值，并根据该值自适应产生色调映像曲线 取值范围：[0x0, 0x800] 数据类型：CVI_U32
HdrStrength	控制 HDR 增强效果参数，数值越大整体通透度拉伸越强，反之则拉伸越弱。 取值范围：[0x0, 0xff] 数据类型：CVI_U16
DarkOffset	控制暗区提亮，数值越大暗区提亮越多，但亮区通透度会降低。不建议将该值设置过大。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMinThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMaxThrd	局部对比增强：用来控制暗区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastDarkMinWeight	局部对比增强：用来控制暗区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastDarkMaxWeight	局部对比增强：用来控制暗区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
ContrastBrightMinThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最小临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMaxThrd	局部对比增强：用来控制亮区对比的权重随动态范围自适应最大临界值。 取值范围：[0x0, 0x3ff] 数据类型：CVI_U16
ContrastBrightMinWeight	局部对比增强：用来控制亮区对比最小临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

下页继续

表 31.2 – 续上页

成员名称	描述
ContrastBrightMaxWeight	局部对比增强：用来控制亮区对比最大临界值下所对应的权重。 取值范围：[0x0, 0xff] 数据类型：CVI_U8
TotalGain	整体的细节强度， $1x = 32$ 取值范围：[0x0, 0xff] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDRCAttr
- CVI\_ISP\_GetDRCAttr

### 31.3.3 ISP\_DRC\_ATTR\_S

**【说明】**

DRC 属性参数

**【定义】**

```
typedef struct ISP_DRC_ATTR_S {
    CVI_BOOL Enable; /*RW; Range:[0, 1]*/
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
    CVI_U8 TuningMode; /*RW; Range:[0x0, 0x7]*/
    CVI_BOOL LocalToneEn; /*RW; Range:[0x0, 0x1]*/
    CVI_BOOL LocalToneRefineEn; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 ToneCurveSelect; /*RW; Range:[0x0, 0x1]*/
    CVI_U16 CurveUserDefine[DRC_GLOBAL_USER_DEFINE_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 DarkUserDefine[DRC_DARK_USER_DEFINE_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 BrightUserDefine[DRC_BRIGHT_USER_DEFINE_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U32 ToneCurveSmooth; /*RW; Range:[0x0, 0x1f4]*/
    CVI_U8 CoarseFiltScale; /*RW; Range:[0x3, 0x6]*/
    CVI_U8 SdrTargetYGainMode; /*RW; Range:[0x0, 0x1]*/
    CVI_BOOL DetailEnhanceEn; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 LumaGain[33]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 DetailEnhanceMtIn[4]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailEnhanceMtOut[4]; /*RW; Range:[0x0, 0x100]*/
    CVI_U8 OverShootThd; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThd; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 OverShootThdMax; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThdMin; /*RW; Range:[0x0, 0xff]*/
    CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/
}
```

(下页继续)

(续上页)

```

CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/
CVI_BOOL dbg_182x_sim_enable; /*RW; Range:[0, 1]*/
CVI_U8 DarkMapStr; /*RW; Range:[0x0, 0x80]*/
CVI_U8 BritMapStr; /*RW; Range:[0x0, 0x80]*/
CVI_U8 SdrDarkMapStr; /*RW; Range:[0x0, 0x80]*/
CVI_U8 SdrBritMapStr; /*RW; Range:[0x0, 0x80]*/
CVI_U32 DRCMu[32]; /*RW; Range:[0x0, 0x7ffffff]*/
ISP_DRC_MANUAL_ATTR_S stManual;
ISP_DRC_AUTO_ATTR_S stAuto;
} ISP_DRC_ATTR_S;

```

## 【成员】

成员名称	描述
Enable	DRC 模块使能 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
TuningMode	调整模式，输出可视化辅助信息，帮助用户调试 0: 不输出可视化辅助信息 1: 输出 SE 画面亮部信息可视化结果。 2: 输出 LE 画面暗部信息可视化结果。 3: 输出 SE 画面 DC 信息可视化结果。 4: 输出 LE 画面 DC 信息可视化结果 取值范围：[0x0, 0x7] 数据类型：CVI_U8
LocalToneEn	Bright Local Tone Mapping 及 Dark Local Tone Mapping 的使能 取值范围：[0x0, 0x1] 数据类型：CVI_BOOL
LocalToneRefineEn	优化 local tone mapping 的使能，让区域分的更精细 取值范围：[0x0, 0x1] 数据类型：CVI_BOOL
CoarseFltScale	控制滤波窗口大小，相比 DRangeFltScale 调节步长更大。值越大滤波窗口越大，细节表现越丰富。反之则越弱 取值范围：[0x3, 0x6] 数据类型：CVI_U8
DarkUserDefine[257]	用户自定义暗部色调映像曲线 取值范围：[0x0, 0xffff] 数据类型：CVI_U16
BrightUserDefine[513]	用户自定义亮部色调映像曲线 取值范围：[0x0, 0xffff] 数据类型：CVI_U16

下页继续

表 31.3 – 续上页

成员名称	描述
CurveUserDefine[1025]	用户自定义曲线 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
ToneCurveSelect	Tone Curve 曲线选择: 0: 选择用户自定义曲线。 1: 选择 Asymmetry 曲线 取值范围: [0x0, 0x1] 数据类型: CVI_U8
LinearStart	在指定的值域范围内限制色调映色压缩的强度, 能保留影像值落于该区域内的通透度。LinearStart 定义该值域的起始点值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
LinearEnd	在指定的值域范围内限制色调映色压缩的强度, 能保留影像值落于该区域内的通透度。LinearEnd 定义该值域的终点值 取值范围: [0x0, 0xffff] 数据类型: CVI_U16
ToneCurveSmooth	Tone Curve 曲线时间域上变化的平顺度。数值越大时间域变化越平顺, 反之则变化越快 取值范围: [0x0, 0x1f4] 数据类型: CVI_U32
DetailEnhanceEnable	Sharpen 模块使能 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
DetailEnhanceEn	ltn_ee 模块使能, 加强 HDR 的细节 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
LumaGain[33]	根据亮度调整加强的细节强度, 1x = 64 取值范围: [0x0, 0xff] 数据类型: CVI_U8
DetailEnhanceMtIn[4]	根据 motion 的强度, 调整加强的细节 取值范围: [0x0, 0xff] 数据类型: CVI_U8
DetailEnhanceMtOut[4]	根据 motion 的强度, 调整加强的细节 取值范围: [0x0, 0x100] 数据类型: CVI_U16
OverShootThd	白边锐化上限幅度 取值范围: [0x0, 0xff] 数据类型: CVI_U8
UnderShootThd	黑边锐化上限幅度 取值范围: [0x0, 0xff] 数据类型: CVI_U8
OverShootGain	白边锐化的强度, 16 为一倍 取值范围: [0x0, 0x3f] 数据类型: CVI_U8
UnderShootGain	黑边锐化的强度, 16 为一倍 取值范围: [0x0, 0x3f] 数据类型: CVI_U8

下页继续

表 31.3 – 续上页

成员名称	描述
OverShootThdMax	白边锐化最大上限幅度 取值范围: [0x0, 0xff] 数据类型: CVI_U8
UnderShootThdMin	黑边锐化最大上限幅度 取值范围: [0x0, 0xff] 数据类型: CVI_U8
SoftClampEnable	平滑的处理边缘加强 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
SoftClampUB	平滑处理边缘加强的上下界, 设定的值越大, 则边缘加强的越连续, 但加强的效果也越弱 取值范围: [0x0, 0xff] 数据类型: CVI_U8
SoftClampLB	平滑处理边缘加强的上下界, 设定的值越大, 则边缘加强的越连续, 但加强的效果也越弱 取值范围: [0x0, 0xff] 数据类型: CVI_U8
dbg_182x_sim_enable	debug mode for 182x performance simulation 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
DrcMode	DRC Mode 0: 在 HSV domain 做 DRC 1: 在 RGB domain 做 DRC 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
SatEnable	饱和度调试功能使能 取值范围: [0x0, 0x1] 数据类型: CVI_BOOL
SdrToneCurveSelect	SDR DRC Tone curve 曲线选择 0: Bypass Mode 4: SdrDarkEnhance(SdrDE) Mode 取值范围: [0x0, 0x4] 数据类型: CVI_U8
DEAdaptEn	Dark Tone 自适应功能使能 取值范围: [0, 1] 数据类型: CVI_BOOL
DEAdaptMode	Dark Tone 自适应功能模式 0: 不考虑 Dark info 1: 考虑 Dark info 取值范围: [0, 1] 数据类型: CVI_BOOL
SdrDEAdaptMode	Dark Tone 自适应功能模式 0: 不考虑 Dark info 1: 考虑 Dark info 取值范围: [0, 1] 数据类型: CVI_BOOL

下页继续

表 31.3 – 续上页

成员名称	描述
DarkMapStr	Dark Tone 作用的强度，建议采用默认值，值越高 dark tone 的影响越强 取值范围：[0x1, 0x80] 数据类型：CVI_U8
SdrDarkMapStr	Dark Tone 作用的强度，建议采用默认值，值越高 dark tone 的影响越强 取值范围：[0x1, 0x80] 数据类型：CVI_U8
BritMapStr	Bright Tone 作用的强度，建议采用默认值，值越高 bright tone 的影响越强 取值范围：[0x1, 0x80] 数据类型：CVI_U8
SdrBritMapStr	Bright Tone 作用的强度，建议采用默认值，值越高 bright tone 的影响越强 取值范围：[0x1, 0x80] 数据类型：CVI_U8
SdrTargetYGainMode	SdrTargetY Gain Mode 开关 0: 直接指定画面平均的目标亮度 LmapLeMode 1: 基于画面的平均，拉亮倍数 1x=32, 2x=64 取值范围：[0x0, 0x1] 数据类型：CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetDRCAAttr
- CVI\_ISP\_GetDRCAAttr



# 32 DIS

## 32.1 功能描述

DIS 模块通过比较当前图像以及前一帧图像来计算出当前的位移量，并将其传递至后级模块进行图像裁切，进而实现防抖动的效果。

## 32.2 API 参考

- CVI\_ISP\_SetDISAttr：设置 DIS 属性参数
- CVI\_ISP\_GetDISAttr：获取 DIS 属性参数
- CVI\_ISP\_SetDISConfig：设置 DIS 控制参数
- CVI\_ISP\_GetDISConfig：获取 DIS 控制参数

### 32.2.1 CVI\_ISP\_SetDISAttr

#### 【描述】

设置 DIS 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetDisAttr(VI_PIPE ViPipe, const ISP_DIS_ATTR_S *pstDisAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDisAttr	DIS 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。



**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetDISAttr](#)

## 32.2.2 CVI\_ISP\_GetDISAttr

**【描述】**

获取 DIS 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDisAttr(VI_PIPE ViPipe, ISP_DIS_ATTR_S *pstDisAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDisAttr	DIS 属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetDISAttr](#)

### 32.2.3 CVI\_ISP\_SetDISConfig

**【描述】**

获取 DIS 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_SetDisConfig(VI_PIPE ViPipe, const ISP_DIS_CONFIG_S *pstDisConfig);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDisConfig	DIS 控制参数	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetDISConfig](#)

### 32.2.4 CVI\_ISP\_GetDISConfig

**【描述】**

获取 DIS 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetDisConfig(VI_PIPE ViPipe, const ISP_DIS_CONFIG_S *pstDisConfig);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstDisConfig	DIS 控制参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetDISConfig](#)

## 32.3 数据类型

- `ISP_DIS_ATTR_S` : DIS 属性参数
- `ISP_DIS_CONFIG_S` : DIS 设定参数
- `DIS_MODE_E` : DIS 模式参数
- `DIS_MOTION_LEVEL_E` : DIS 级别参数

### 32.3.1 ISP\_DIS\_ATTR\_S

**【说明】**

DIS 属性参数

**【定义】**

```
typedef struct _ISP_DIS_ATTR_S {  
    CVI_BOOL enable;  
    CVI_U32 movingSubjectLevel;  
    CVI_U32 horizontalLimit;  
    CVI_U32 verticalLimit;  
}
```

(下页继续)

(续上页)

```

    CVI_BOOL stillCrop;
} ISP_DIS_ATTR_S;

```

**【成员】**

成员名称	描述
enable	DIS 使能开关 CVI_FALSE: 不使能 CVI_TRUE: 使能
movingSubjectLevel	用于判断物体是否移动的参数, 值越小防抖效果越好, 但容易出现偏移, 反之抖动容易变大, 但减少偏移现象。 取值范围: [0, 6] 数据类型: CVI_U32
horizontalLimit	水平方向的位移限制, 当大面积物体位移造成背景位移达到一定幅度, 则不做防抖效果。计算方式为 $\text{image\_width} * \text{horizontalLimit} / 1000$ 取值范围: [0, 0x3E8] 数据类型: CVI_U32
verticalLimit	垂直方向的位移限制, 当大面积物体位移造成背景位移达到一定幅度, 则不做防抖效果。计算方式为 $\text{image\_height} * \text{verticalLimit} / 1000$ 取值范围: [0, 0x3E8] 数据类型: CVI_U32
stillCrop	关闭 DIS 防抖效果, 但影像保持裁剪比例输出, 避免影像大小变化 CVI_FALSE: 不使能 CVI_TRUE: 使能

**【注意事项】**

无。

**【相关数据类型及接口】**

- [CVI\\_ISP\\_SetDISAttr](#)
- [CVI\\_ISP\\_GetDISAttr](#)

### 32.3.2 ISP\_DIS\_CONFIG\_S

**【说明】**

DIS 设定参数

**【定义】**

```

typedef struct _ISP_DIS_CONFIG_S {
    DIS_MODE_E mode;
    DIS_MOTION_LEVEL_E motionLevel;
    CVI_U32 cropRatio;
} ISP_DIS_CONFIG_S;

```

## 【成员】

成员名称	描述
mode	数字防抖动算法模式，目前只支持一种算法模式 取值范围：[0, 0] 数据类型：DIS_MODE_E
motionLevel	Camera 的运动级别，目前只支持 DIS_MOTION_LEVEL_NORMAL 取值范围：[1, 1] 数据类型：DIS_MOTION_LEVEL
cropRatio	DIS 输出的影像裁切比例，数值越大则保留的影像越大，反之则越小。建议默认值为 94 取值范围：[50, 98] 数据类型：CVI_U32

## 【注意事项】

cropRatio 会影响 DIS 算法能支持的最大位移程度，假设 CropRatio 为 98，表示只剩下 2% 的宽高能做位移

## 【相关数据类型及接口】

- CVI\_ISP\_SetDISConfig
- CVI\_ISP\_GetDISConfig

### 32.3.3 DIS\_MODE\_E

## 【说明】

定义 DIS 防抖算法模式

## 【定义】

```
typedef enum _ISP_DIS_MODE_E {
    DIS_MODE_2_DOF_GME = 0, /* Only use with GME in 2 dof */
    DIS_MODE_DEBUG,
    DIS_MODE_DOF_BUTT,
} DIS_MODE_E;
```

## 【成员】

成员名称	描述
DIS_MODE_2_DOF_GME	二自由度 GME 算法
DIS_MODE_DEBUG	调试模式

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetDISConfig

- CVI\_ISP\_GetDISConfig

### 32.3.4 DIS\_MOTION\_LEVEL\_E

#### 【说明】

定义镜头的运动级别

#### 【定义】

```
typedef enum _ISP_DIS_MOTION_LEVEL_E {  
    DIS_MOTION_LEVEL_NORMAL = 1,  
    DIS_MOTION_LEVEL_BUTT  
} DIS_MOTION_LEVEL_E;
```

#### 【成员】

成员名称	描述
DIS_MOTION_LEVEL_NORMAL	正常级别的镜头晃动幅度

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- CVI\_ISP\_SetDISConfig
- CVI\_ISP\_GetDISConfig

# 33 Mono

## 33.1 功能描述

本章节描述如何设置单色模式与属性。

## 33.2 API 参考

- `CVI_ISP_SetMonoAttr`：设置 Mono 属性参数
- `CVI_ISP_GetMonoAttr`：获取 Mono 属性参数

### 33.2.1 CVI\_ISP\_SetMonoAttr

#### 【描述】

设置 Mono 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetMonoAttr(VI_PIPE ViPipe, const ISP_MONO_ATTR_S *pstMonoAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMonoAttr	Mono 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`

· 库文件: libisp.so

【注意】

无。

【举例】

无。

【相关主题】

· [CVI\\_ISP\\_GetMonoAttr](#)

### 33.2.2 CVI\_ISP\_GetMonoAttr

【描述】

获取 Mono 属性参数

【语法】

```
CVI_S32 CVI_ISP_GetMonoAttr(VI_PIPE ViPipe, *pstMonoAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstMonoAttr	Mono 属性参数	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

【注意】

无。

【举例】

无。

【相关主题】

· [CVI\\_ISP\\_SetMonoAttr](#)



## 33.3 数据类型

- `ISP_MONO_ATTR_S` : Mono 属性参数

### 33.3.1 `ISP_MONO_ATTR_S`

#### 【说明】

Mono 属性参数

#### 【定义】

```
typedef struct _ISP_MONO_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_U8 UpdateInterval;  
} ISP_MONO_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	单色模式使能。 取值范围：[0x0, 0x1] 数据类型：CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- `CVI_ISP_SetMonoAttr`
- `CVI_ISP_GetMonoAttr`

# 34 YCONTRAST

## 34.1 功能描述

Y 值域线性对比度。

## 34.2 API 参考

- `CVI_ISP_SetYContrastAttr`：设置 Y 值域对比属性参数
- `CVI_ISP_GetYContrastAttr`：获取 Y 值域对比属性参数

### 34.2.1 CVI\_ISP\_SetYContrastAttr

#### 【描述】

设置 Y 值域对比属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetYContrastAttr(VI_PIPE ViPipe, const ISP_YCONTRAST_ATTR_S *  
↪pstYContrastAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYContrastAttr	Y 值域对比属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_GetYContrastAttr](#)

### 34.2.2 CVI\_ISP\_GetYContrastAttr

【描述】

获取 Y 值域对比属性参数

【语法】

```
CVI_S32 CVI_ISP_GetYContrastAttr(VI_PIPE ViPipe, ISP_YCONTRAST_ATTR_S *  
↪pstYContrastAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstYContrastAttr	伽玛属性参数	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_SetYContrastAttr](#)

## 34.3 数据类型

- `ISP_YCONTRAST_MANUAL_ATTR_S`：伽玛属性手动参数
- `ISP_YCONTRAST_AUTO_ATTR_S`：伽玛属性自动参数
- `ISP_YCONTRAST_ATTR_S`：伽玛属性参数

### 34.3.1 `ISP_YCONTRAST_MANUAL_ATTR_S`

#### 【说明】

Y 值域对比属性手动参数

#### 【定义】

```
typedef struct _ISP_YCONTRAST_MANUAL_ATTR_S {  
    CVI_U8 ContrastLow;  
    CVI_U8 ContrastHigh;  
    CVI_U8 CenterLuma;  
} ISP_YCONTRAST_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
ContrastLow	小于中心点位置的区域对比度强度。值越大，对比越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
ContrastHigh	大于中心点位置的区域对比度强度。值越大，对比越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
CenterLuma	中心点位置，会以中心点往两边加强对比。 取值范围：[0x0, 0x40] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- `CVI_ISP_SetYContrastAttr`
- `CVI_ISP_GetYContrastAttr`

### 34.3.2 ISP\_YCONTRAST\_AUTO\_ATTR\_S

**【说明】**

Y 值域对比属性自动参数

**【定义】**

```
typedef struct _ISP_YCONTRAST_AUTO_ATTR_S {  
    CVI_U8 ContrastLow[ISP_AUTO_LV_NUM];  
    CVI_U8 ContrastHigh[ISP_AUTO_LV_NUM];  
    CVI_U8 CenterLuma[ISP_AUTO_LV_NUM];  
} ISP_YCONTRAST_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
ContrastLow	小于中心点位置的区域对比度强度。值越大，对比越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
ContrastHigh	大于中心点位置的区域对比度强度。值越大，对比越强。 取值范围：[0x0, 0x64] 数据类型：CVI_U8
CenterLuma	中心点位置，会以中心点往两边加强对比。 取值范围：[0x0, 0x40] 数据类型：CVI_U8

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetYContrastAttr
- CVI\_ISP\_GetYContrastAttr

### 34.3.3 ISP\_YCONTRAST\_ATTR\_S

**【说明】**

Y 值域对比属性参数

**【定义】**

```
typedef struct _ISP_YCONTRAST_ATTR_S {  
    CVI_BOOL Enable;  
    OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_YCONTRAST_MANUAL_ATTR_S stManual;  
    ISP_YCONTRAST_AUTO_ATTR_S stAuto;  
} ISP_YCONTRAST_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	YCONTRAST 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
enOpType	工作类型 OP_TYPE_AUTO: 自动模式 OP_TYPE_MANUAL: 手动模式
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetYContrastAttr
- CVI\_ISP\_GetYContrastAttr

# 35 CA

## 35.1 功能描述

调整 UV domain 饱和度, 此模组分为 CA 模式与 CP (热成像) 模式, 同时间只能选一个开启。

CA 模式: 根据 input Y 来调整饱和度, 因此可以局部调整饱和度。

CP 模式: 因为热成像模式只有 Y 值, 会根据 Y 值查找预先调试的颜色模板, 查找对应的 YUV 值, 使其上色。

## 35.2 API 参考

- `CVI_ISP_SetCAAttr`: 设置饱和度属性参数
- `CVI_ISP_GetCAAttr`: 获取饱和度属性参数

### 35.2.1 CVI\_ISP\_SetCAAttr

#### 【描述】

设置饱和度属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCAAttr(VI_PIPE ViPipe, const ISP_CA_ATTR_S *pstCAAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCAAttr	饱和度属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetCAAttr](#)

## 35.2.2 CVI\_ISP\_GetCAAttr

**【描述】**

获取饱和度属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCAAttr(VI_PIPE ViPipe, ISP_CA_ATTR_S *pstCAAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCAAttr	饱和度属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**



无。

【相关主题】

- [CVI\\_ISP\\_SetCAAttr](#)

35.3 数据类型

- [ISP\\_CA\\_MANUAL\\_ATTR\\_S](#) : 饱和度属性手动参数
- [ISP\\_CA\\_AUTO\\_ATTR\\_S](#) : 饱和度属性自动参数
- [ISP\\_CA\\_ATTR\\_S](#) : 饱和度属性参数

35.3.1 ISP\_CA\_MANUAL\_ATTR\_S

【说明】

饱和度属性手动参数

【定义】

```
typedef struct ISP_CA_MANUAL_ATTR_S {
    CVI_U16 ISORatio;
    CVI_U16 YRatioLut[CA_LUT_NUM];
} ISP_CA_MANUAL_ATTR_S;
```

【成员】

成员名称	描述
ISORatio	CA 模式, 根据 ISO 值查找 UV 的增益。所有像素点的 UV 调整增益都是相同的, 建议在低 ISO 的时候此增益可以设置大一些, 而高 ISO 时此增益值可以设定小一些, 来抑制暗区的色噪。 取值范围: [0x0, 0x7ff] 数据类型: CVI_U16
YRatioLut[CA_LUT_NUM]	CA 模式, 根据亮度 Y 查找 UV 的增益。此值根据不同亮度等级可设置不同的 UV 增益, 建议在亮区的增益可以设置大一些, 颜色会较为鲜艳, 而暗区的增益可以设置小一些, 来抑制暗区色噪。 取值范围: [0x0, 0x7ff] 数据类型: CVI_U16

【注意事项】

无。

【相关数据类型及接口】

- [CVI\\_ISP\\_SetCAAttr](#)
- [CVI\\_ISP\\_GetCAAttr](#)

### 35.3.2 ISP\_CA\_AUTO\_ATTR\_S

#### 【说明】

饱和度属性自动参数

#### 【定义】

```
typedef struct _ISP_CA2_AUTO_ATTR_S {
    CVI_U16 ISORatio[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 YRatioLut[CA_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CA2_AUTO_ATTR_S;
```

#### 【成员】

成员名称	描述
ISORatio	CA 模式, 根据 ISO 值查找 UV 的增益。所有像素点的 UV 调整增益都是相同的, 建议在低 ISO 的时候此增益可以设置大一些, 而高 ISO 时此增益值可以设定小一些, 来抑制暗区的色噪。 取值范围: [0x0, 0x7ff] 数据类型: CVI_U16
YRatioLut[CA_LUT_NUM]	CA 模式, 根据亮度 Y 查找 UV 的增益。此值根据不同亮度等级可设置不同的 UV 增益, 建议在亮区的增益可以设置大一些, 颜色会较为鲜艳, 而暗区的增益可以设置小一些, 来抑制暗区色噪。 取值范围: [0x0, 0x7ff] 数据类型: CVI_U16

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- CVI\_ISP\_SetCAAttr
- CVI\_ISP\_GetCAAttr

### 35.3.3 ISP\_CA\_ATTR\_S

#### 【说明】

饱和度属性参数

#### 【定义】

```
typedef struct _ISP_CA_ATTR_S {
    CVI_BOOL Enable;
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval;
    CVI_BOOL CaCpMode;
    CVI_U8 CPLutY[CA_LUT_NUM];
    CVI_U8 CPLutU[CA_LUT_NUM];
```

(下页继续)

(续上页)

```

CVI_U8 CPLutV[CA_LUT_NUM];
ISP_CA_MANUAL_ATTR_S stManual;
ISP_CA_AUTO_ATTR_S stAuto;
} ISP_CA_ATTR_S;

```

**【成员】**

成员名称	描述
Enable	CA 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
CaCpMode	CA 模式选择: 0: CA 模式。 1: CP 模式。 取值范围: [0, 1] 数据类型: CVI_BOOL
CPLutY	CP 模式, 根据亮度 Y 查找 LUT 对应的 Y 值。 取值范围: [0, 0xff] 数据类型: CVI_U8
CPLutU	CP 模式, 根据亮度 Y 查找 LUT 对应的 U 值。 取值范围: [0, 0xff] 数据类型: CVI_U8
CPLutV	CP 模式, 根据亮度 Y 查找 LUT 对应的 V 值。 取值范围: [0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCA2Attr
- CVI\_ISP\_GetCA2Attr

# 36 CA2

## 36.1 功能描述

调整 UV domain 饱和度, 根据输入像素的饱和度, 重新调整饱和度等级

## 36.2 API 参考

- `CVI_ISP_SetCA2Attr` : 设置饱和度属性参数
- `CVI_ISP_GetCA2Attr` : 获取饱和度属性参数

### 36.2.1 CVI\_ISP\_SetCA2Attr

#### 【描述】

设置饱和度属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCA2Attr(VI_PIPE ViPipe, const ISP_CA2_ATTR_S *pstCA2Attr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCA2Attr	饱和度属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`

- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetCA2Attr](#)

## 36.2.2 CVI\_ISP\_GetCA2Attr

**【描述】**

获取饱和度属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCA2Attr(VI_PIPE ViPipe, ISP_CA2_ATTR_S *pstCA2Attr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCA2Attr	饱和度属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetCA2Attr](#)

## 36.3 数据类型

- `ISP_CA2_MANUAL_ATTR_S`：饱和度属性手动参数
- `ISP_CA2_AUTO_ATTR_S`：饱和度属性自动参数
- `ISP_CA2_ATTR_S`：饱和度属性参数

### 36.3.1 `ISP_CA2_MANUAL_ATTR_S`

#### 【说明】

饱和度属性手动参数

#### 【定义】

```
typedef struct _ISP_CA2_MANUAL_ATTR_S {  
    CVI_U16 Ca2In[CA_LITE_NODE];  
    CVI_U16 Ca2Out[CA_LITE_NODE];  
} ISP_CA2_MANUAL_ATTR_S;
```

#### 【成员】

成员名称	描述
<code>Ca2In[CA_LITE_NODE]</code>	由六个数值组成的数组，决定输入饱和度等级 取值范围：[0x0, 0xc0] 数据类型：CVI_U8
<code>Ca2Out[CA_LITE_NODE]</code>	由六个数值组成的数组，定义输出的 UV 增益。根据输入饱和度查找 UV 的增益，值越大，饱和度越高；反之，则越小 取值范围：[0x0, 0x7ff] 数据类型：CVI_U16

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- `CVI_ISP_SetCA2Attr`
- `CVI_ISP_GetCA2Attr`

### 36.3.2 ISP\_CA2\_AUTO\_ATTR\_S

**【说明】**

饱和度属性自动参数

**【定义】**

```
typedef struct _ISP_CA2_AUTO_ATTR_S {  
    CVI_U16 Ca2In[CA_LITE_NODE][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 Ca2Out[CA_LITE_NODE][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CA2_AUTO_ATTR_S;
```

**【成员】**

成员名称	描述
Ca2In[CA_LITE_NODE]	由六个数值组成的数组，决定输入饱和度等级 取值范围：[0x0, 0xc0] 数据类型：CVI_U8
Ca2Out[CA_LITE_NODE]	由六个数值组成的数组，定义输出的 UV 增益。根据输入饱和度查找 UV 的增益，值越大，饱和度越高；反之，则越小 取值范围：[0x0, 0x7ff] 数据类型：CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCA2Attr
- CVI\_ISP\_GetCA2Attr

### 36.3.3 ISP\_CA2\_ATTR\_S

**【说明】**

饱和度属性参数

**【定义】**

```
typedef struct _ISP_CA2_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_OP_TYPE_E enOpType;  
    CVI_U8 UpdateInterval;  
    ISP_CA2_MANUAL_ATTR_S stManual;  
    ISP_CA2_AUTO_ATTR_S stAuto;  
} ISP_CA2_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	CA2 模块使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好 取值范围: [0x0, 0xff] 数据类型: CVI_U8
stManual	手动模式参数属性
stAuto	自动模式参数属性

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCA2Attr
- CVI\_ISP\_GetCA2Attr



# 37 CLUT

## 37.1 功能描述

透过一个 17x17x17 的 3D LUT, 在 RGB domain 上做线性转换。将 RGB 像素值查表内插得到新的 RGB 像素值, 可以借此调试颜色、亮度。

## 37.2 API 参考

- `CVI_ISP_SetClutAttr` : 设置 CLUT 属性参数
- `CVI_ISP_GetClutAttr` : 获取 CLUT 属性参数
- `CVI_ISP_SetClutSaturationAttr` : 设置 CLUT SbyS 属性参数
- `CVI_ISP_GetClutSaturationAttr` : 获取 CLUT SbyS 属性参数

### 37.2.1 CVI\_ISP\_SetClutAttr

#### 【描述】

设置 CLUT 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetClutAttr(VI_PIPE ViPipe, const ISP_CLUT_ATTR_S *pstClutAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCLUTAttr	CLUT 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败, 其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_GetClutAttr](#)

### 37.2.2 CVI\_ISP\_GetClutAttr

【描述】

获取 CLUT 属性参数

【语法】

```
CVI_S32 CVI_ISP_GetClutAttr(VI_PIPE ViPipe, ISP_CLUT_ATTR_S *pstClutAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCLUTAttr	CLUT 属性参数	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

【注意】

无。

【举例】

无。

【相关主题】

- [CVI\\_ISP\\_SetClutAttr](#)

### 37.2.3 CVI\_ISP\_SetClutSaturationAttr

**【描述】**

设置 CLUT SbyS 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_SetClutSaturationAttr(VI_PIPE ViPipe, ISP_CLUT_SATURATION_ATTR_S_
→ *pstClutSaturationAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstClutSaturationAttr	CLUT SbyS 属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- CVI\_ISP\_GetClutSaturationAttr

### 37.2.4 CVI\_ISP\_GetClutSaturationAttr

**【描述】**

获取 CLUT SbyS 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetClutSaturationAttr(VI_PIPE ViPipe, ISP_CLUT_SATURATION_ATTR_
→ S *pstClutSaturationAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstClutSaturationAttr	CLUT SbyS 属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetClutSaturationAttr](#)

## 37.3 数据类型

- `ISP_CLUT_ATTR_S` : CLUT 属性参数
- `ISP_CLUT_SATURATION_MANUAL_ATTR_S` : CLUT SbyS 属性手动参数
- `ISP_CLUT_SATURATION_AUTO_ATTR_S` : CLUT SbyS 属性自动参数
- `ISP_CLUT_SATURATION_ATTR_S` : CLUT SbyS 属性参数

### 37.3.1 ISP\_CLUT\_ATTR\_S

**【说明】**

CLUT 属性参数

**【定义】**

```
typedef struct _ISP_CLUT_ATTR_S {  
    CVI_BOOL Enable;  
    CVI_U8 UpdateInterval;  
    CVI_U16 ClutR[ISP_CLUT_LUT_LENGTH];  
    CVI_U16 ClutG[ISP_CLUT_LUT_LENGTH];  
    CVI_U16 ClutB[ISP_CLUT_LUT_LENGTH];  
} ISP_CLUT_ATTR_S;
```

**【成员】**

成员名称	描述
Enable	CLUT 模块使能。 0: 关闭。 1: 使能。 取值范围: [0, 1] 数据类型: CVI_BOOL
UpdateInterval	影响参数更新间隔, 值越大画面变化越慢, 效能越好。 取值范围: [0x0, 0xff] 数据类型: CVI_U8
ClutR	R 通道的 LUT。 取值范围: [0x0, 0x3ff] 数据类型: CVI_U16
ClutG	G 通道的 LUT。 取值范围: [0x0, 0x3ff] 数据类型: CVI_U16
ClutB	B 通道的 LUT。 取值范围: [0x0, 0x3ff] 数据类型: CVI_U16

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetClutAttr
- CVI\_ISP\_GetClutAttr

### 37.3.2 ISP\_CLUT\_SATURATION\_MANUAL\_ATTR\_S

**【说明】**

CLUT SbyS 属性手动参数

**【定义】**

```
typedef struct _ISP_CLUT_SATURATION_MANUAL_ATTR_S {  
    CVI_U16 SatIn[4];  
    CVI_U16 SatOut[4];  
} ISP_CLUT_SATURATION_MANUAL_ATTR_S;
```

## 【成员】

成员名称	描述
SatIn [4]	SbyS 输入 取值范围：[0, 0x2000] 数据类型：CVI_U16
SatOut [4]	SbyS 输出 取值范围：[0, 0x2000] 数据类型：CVI_U16

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetClutSaturationAttr
- CVI\_ISP\_GetClutSaturationAttr

### 37.3.3 ISP\_CLUT\_SATURATION\_AUTO\_ATTR\_S

## 【说明】

CLUT SbyS 属性手动参数

## 【定义】

```
typedef struct _ISP_CLUT_SATURATION_AUTO_ATTR_S {  
    CVI_U16 SatIn[4][ISP_AUTO_ISO_STRENGTH_NUM];  
    CVI_U16 SatOut[4][ISP_AUTO_ISO_STRENGTH_NUM];  
} ISP_CLUT_SATURATION_AUTO_ATTR_S;
```

## 【成员】

成员名称	描述
SatIn [4]	SbyS 输入 取值范围：[0, 0x2000] 数据类型：CVI_U16
SatOut [4]	SbyS 输出 取值范围：[0, 0x2000] 数据类型：CVI_U16

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetClutSaturationAttr
- CVI\_ISP\_GetClutSaturationAttr

37.3.4 ISP\_CLUT\_SATURATION\_ATTR\_S

【说明】

CLUT SbyS 属性手动参数

【定义】

```
typedef struct _ISP_CLUT_SATURATION_ATTR_S {
    CVI_BOOL Enable;
    ISP_OP_TYPE_E enOpType;
    ISP_CLUT_SATURATION_MANUAL_ATTR_S stManual;
    ISP_CLUT_SATURATION_AUTO_ATTR_S stAuto;
} ISP_CLUT_SATURATION_ATTR_S;
```

【成员】

成员名称	描述
Enable	SbyS 模块使能。 0: 关闭。 1: 使能。 取值范围：[0, 1] 数据类型：CVI_BOOL
enOpType	选择手动或自动模式
stManual	手动模式参数属性
stAuto	自动模式参数属性

【注意事项】

无。

【相关数据类型及接口】

- CVI\_ISP\_SetClutSaturationAttr
- CVI\_ISP\_GetClutSaturationAttr

# 38 CSC

## 38.1 功能描述

该功能提供了色域转换时的一些相关画质设定，通过设定 hue、luma、saturation、contrast、色域转换标准规范类型，进而影响画面的效果，而无需知晓转换矩阵的计算过程，当然也可以让使用者直接通过设定转换矩阵的 coeff 和 offset，进而影响画质效果

## 38.2 API 参考

- CVI\_ISP\_SetCSCAttr：设置 CSC 属性参数
- CVI\_ISP\_GetCSCAttr：获取 CSC 属性参数

### 38.2.1 CVI\_ISP\_SetCSCAttr

#### 【描述】

设置 CSC 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetCSCAttr(VI_PIPE ViPipe, const ISP_CSC_ATTR_S *pstCSCAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCSCAttr	CSC 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。



**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_GetCSCAttr](#)

## 38.2.2 CVI\_ISP\_GetCSCAttr

**【描述】**

获取 CSC 属性参数

**【语法】**

```
CVI_S32 CVI_ISP_GetCSCAttr(VI_PIPE ViPipe, ISP_CSC_ATTR_S *pstCSCAttr);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstCSCAttr	CSC 属性参数	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

- [CVI\\_ISP\\_SetCSCAttr](#)

## 38.3 数据类型

- `ISP_CSC_ATTR_S` : CSC 属性参数
- `ISP_CSC_COLORGAMUT` : CSC 色域转换标准格式
- `ISP_CSC_MATRX_S` : CSC 自定义转换矩阵

### 38.3.1 `ISP_CSC_ATTR_S`

#### 【说明】

CSC 属性参数

#### 【定义】

```
typedef struct _ISP_CSC_ATTR_S {  
    CVI_BOOL Enable;  
    ISP_CSC_COLORGAMUT enColorGamut;  
    CVI_U8 UpdateInterval;  
    CVI_U8 Hue;  
    CVI_U8 Luma;  
    CVI_U8 Contrast;  
    CVI_U8 Saturation;  
    ISP_CSC_MATRX_S stUserMatrx;  
} ISP_CSC_ATTR_S;
```

#### 【成员】

成员名称	描述
Enable	CSC 模块使能 取值范围：[0, 1] 数据类型：CVI_BOOL
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
enColorGamut	色域转换类型选择 0: ISP_CSC_COLORGAMUT_BT601 1: ISP_CSC_COLORGAMUT_BT709 2: ISP_CSC_COLORGAMUT_BT2020 3: ISP_CSC_COLORGAMUT_USER
Hue	调节色度 取值范围：[0x0, 0x64] 数据类型：CVI_U8
Luma	调节亮度 取值范围：[0x0, 0x64] 数据类型：CVI_U8
Contrast	调节对比度 取值范围：[0x0, 0x64] 数据类型：CVI_U8
Saturation	调节饱和度 取值范围：[0x0, 0x64] 数据类型：CVI_U8
stUserMatrx	自定义色域转换矩阵

**【注意事项】**

无。

**【相关数据类型及接口】**

- CVI\_ISP\_SetCscAttr
- CVI\_ISP\_GetCscAttr

### 38.3.2 ISP\_CSC\_COLORGAMUT

**【说明】**

色域转换类型选择

**【定义】**

```
typedef enum ISP_CSC_COLORGAMUT {
    ISP_CSC_COLORGAMUT_BT601,
    ISP_CSC_COLORGAMUT_BT709,
    ISP_CSC_COLORGAMUT_BT2020,
    ISP_CSC_COLORGAMUT_USER,
    ISP_CSC_COLORGAMUT_NUM
} ISP_CSC_COLORGAMUT;
```

## 【成员】

成员名称	描述
ISP_CSC_COLORGAMUT_BT601	BT601 色域转换标准
ISP_CSC_COLORGAMUT_BT709	BT709 色域转换标准
ISP_CSC_COLORGAMUT_BT2020	BT2020 色域转换标准
ISP_CSC_COLORGAMUT_USER	自定义色域转换，与ISP_CSC_MATRX_S 联合使用

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetCscAttr
- CVI\_ISP\_GetCscAttr

### 38.3.3 ISP\_CSC\_MATRX\_S

## 【说明】

自定义转换矩阵

## 【定义】

```
typedef struct _ISP_CSC_MATRX_S {
    CVI_S16 userCscCoef[CSC_MATRIX_SIZE];
    CVI_S16 userCscOffset[CSC_OFFSET_SIZE];
} ISP_CSC_MATRX_S;
```

## 【成员】

成员名称	描述
userCsc-Coef[CSC_MATRIX_SIZE]	3*3 色域转换矩阵的系数 取值范围：[-0x2000, 0x1fff] 数据类型：CVI_S16
userCscOffset[CSC_OFFSET_SIZE]	输出 offset 取值范围：[-0x100, 0xff] 数据类型：CVI_S16

## 【注意事项】

无。

## 【相关数据类型及接口】

- CVI\_ISP\_SetCscAttr
- CVI\_ISP\_GetCscAttr

# 39 VC

## 39.1 功能描述

调整 Video Codec 端 Motion Map 设定

## 39.2 API 参考

- `CVI_ISP_SetVCAAttr`：设置 VC 属性参数
- `CVI_ISP_GetVCAAttr`：获取 VC 属性参数

### 39.2.1 CVI\_ISP\_SetVCAAttr

#### 【描述】

设置 VC 属性参数

#### 【语法】

```
CVI_S32 CVI_ISP_SetVCAAttr(VI_PIPE ViPipe, const ISP_VC_ATTR_S *pstVCAAttr);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstVCAAttr	VC 属性参数	输入

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

#### 【需求】

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`

· 库文件: libisp.so

【注意】

无。

【举例】

无。

【相关主题】

· [CVI\\_ISP\\_GetVCAAttr](#)

### 39.2.2 CVI\_ISP\_GetVCAAttr

【描述】

获取 VC 属性参数

【语法】

```
CVI_S32 CVI_ISP_GetVCAAttr(VI_PIPE ViPipe, ISP_VC_ATTR_S *pstVCAAttr);
```

【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstVCAAttr	VC 属性参数	输出

【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

【注意】

无。

【举例】

无。

【相关主题】

· [CVI\\_ISP\\_SetVCAAttr](#)

## 39.3 数据类型

- `ISP_VC_ATTR_S` : VC 属性参数

### 39.3.1 `ISP_VC_ATTR_S`

#### 【说明】

VC 属性参数

#### 【定义】

```
typedef struct _ISP_VC_ATTR_S {  
    CVI_U8 UpdateInterval;  
    CVI_U8 MotionThreshold;  
} ISP_VC_ATTR_S;
```

#### 【成员】

成员名称	输入/输出
UpdateInterval	影响参数更新间隔，值越大画面变化越慢，效能越好 取值范围：[0x0, 0xff] 数据类型：CVI_U8
MotionThreshold	Motion Map 阈值 取值范围：[0x0, 0xff] 数据类型：CVI_U8

#### 【注意事项】

无。

#### 【相关数据类型及接口】

- `CVI_ISP_SetVCAttr`
- `CVI_ISP_GetVCAttr`

# 40 统计讯息

## 40.1 概述

本章节说明 ISP 提供的 3A 统计信息，及配置方式。

## 40.2 API 参考

统计讯息的接口必需在调用 CVI\_ISP\_Init 接口之后才能调用。

- CVI\_ISP\_SetStatisticsConfig：设置 ISP 统计信息配置
- CVI\_ISP\_GetStatisticsConfig：获取 ISP 统计信息配置
- CVI\_ISP\_GetAESTatistics：获取 ISP AE 统计信息
- CVI\_ISP\_GetWBStatistics：获取 ISP AWB 统计信息
- CVI\_ISP\_GetFocusStatistics：获取 ISP AF 统计信息

### 40.2.1 CVI\_ISP\_SetStatisticsConfig

#### 【描述】

设置 ISP 统计讯息配置。

#### 【语法】

```
CVI_S32 CVI_ISP_SetStatisticsConfig(VI_PIPE ViPipe, const ISP_STATISTICS_CFG_S_
→*pstStatCfg);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstStatCfg	ISP 统计信息配置	输入

#### 【返回值】



返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

```
// Set AE / AWB / AF related statistic window setting
VI_PIPE ViPipe = 0;
ISP_STATISTICS_CFG_S stsCfg;

// 设定AE0 windows ROI.
stsCfg.stAECfg.stCrop[0].bEnable = 1;
stsCfg.stAECfg.stCrop[0].u16X = stsCfg.stAECfg.stCrop[0].u16Y = 0;
stsCfg.stAECfg.stCrop[0].u16W = 1920;
stsCfg.stAECfg.stCrop[0].u16H = 1080;

// 设定AWB统计值在x & y方向的window数.
stsCfg.stWBCfg.u16ZoneRow = AWB_ZONE_ORIG_ROW;
stsCfg.stWBCfg.u16ZoneCol = AWB_ZONE_ORIG_COLUMN;

// 设定AWB window ROI.
stsCfg.stWBCfg.stCrop.u16X = stsCfg.stWBCfg.stCrop.u16Y = 0;
stsCfg.stWBCfg.stCrop.u16W = 1920;
stsCfg.stWBCfg.stCrop.u16H = 1080;

// 设定AWB统计亮度的阈值.
stsCfg.stWBCfg.u16BlackLevel = 0;
stsCfg.stWBCfg.u16WhiteLevel = 4095;

// 设定AF统计值开关.
stsCfg.stFocusCfg.stConfig.bEnable = 1;

// 设定AF统计的前处理开关
stsCfg.stFocusCfg.stConfig.stRawCfg.PreGammaEn = 0;
stsCfg.stFocusCfg.stConfig.stPreFltCfg.PreFltEn = 1;

// 设定AF统计值在 x & y方向的window数
stsCfg.stFocusCfg.stConfig.u16Hwnd = 17;
stsCfg.stFocusCfg.stConfig.u16Vwnd = 15;

// 设定AF window ROI.
stsCfg.stFocusCfg.stConfig.stCrop.bEnable = 1;
stsCfg.stFocusCfg.stConfig.stCrop.u16X = AF_XOFFSET_MIN;
stsCfg.stFocusCfg.stConfig.stCrop.u16Y = AF_YOFFSET_MIN;
stsCfg.stFocusCfg.stConfig.stCrop.u16W = 1920 - AF_XOFFSET_MIN * 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16H = 1080 - AF_YOFFSET_MIN * 2;
```

(下页继续)

(续上页)

```
// 设定AF水平方向低通滤波器系数.
stsCfg.stFocusCfg.stConfig.u8HFfltShift = 1;
stsCfg.stFocusCfg.stConfig.s8HFfltLpCoeff[0] = 1;
stsCfg.stFocusCfg.stConfig.s8HFfltLpCoeff[1] = 2;
stsCfg.stFocusCfg.stConfig.s8HFfltLpCoeff[2] = 3;
stsCfg.stFocusCfg.stConfig.s8HFfltLpCoeff[3] = 5;
stsCfg.stFocusCfg.stConfig.s8HFfltLpCoeff[4] = 10;

// 设定AF水平方向高通滤波器系数.
stsCfg.stFocusCfg.stHParam_FIR0.s8HFfltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFfltHpCoeff[1] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFfltHpCoeff[2] = 13;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFfltHpCoeff[3] = 24;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFfltHpCoeff[4] = 0;

// 设定AF垂直方向高通滤波器系数.
stsCfg.stFocusCfg.stVParam_FIR.s8VFfltHpCoeff[0] = 13;
stsCfg.stFocusCfg.stVParam_FIR.s8VFfltHpCoeff[1] = 24;
stsCfg.stFocusCfg.stVParam_FIR.s8VFfltHpCoeff[2] = 0;

// 设定各统计值取值开关.
stsCfg.unKey.bit1FEAeGloStat = stsCfg.unKey.bit1FEAeLocStat =
stsCfg.unKey.bit1AwbStat1 = stsCfg.unKey.bit1AwbStat2 = stsCfg.unKey.bit1FEAfStat = 1;
CVI_ISP_SetStatisticsConfig(ViPipe, &stsCfg);

// 设定AF低通滤波器系数规则.
AF低通滤波器拥有九个参数 {x0, x1, x2, x3, x4, x5, x6, x7, x8} ,
最后四个参数与前四个参数对称(x0 = x8, x1 = x7, x2 = x6, x3 = x5) ,
使用者只须设置前五个参数(x0 ~ x5)即可
// 设定AF水平高通滤波器系数规则.
AF水平高通滤波器拥有九个参数 {x0, x1, x2, x3, x4, x5, x6, x7, x8} ,
最后四个参数与前四个参数 ,
数字相同但正负符号相反 (x0 = -x8, x1 = -x7, x2 = -x6, x3 = -x5) ,
使用者只须设置前五个参数(x0 ~ x5)即可
// 设定AF垂直高通滤波器系数规则.
AF水平高通滤波器拥有五个参数 {x0, x1, x2, x3, x4} ,
最后两个参数与前两个参数 ,
数字相同但正负符号相反 (x0 = -x4, x1 = -x3) ,
使用者只须设置前三个参数(x0 ~ x2)即可
```

filter 参数设定范例

滤波器可通过的频带	滤波器参数 x0	滤波器参数 x1	滤波器参数 x2
0.1 ~ 0.2	20	16	0
0.1 ~ 0.5	17	20	0
0.1 ~ 0.6	13	24	0
0.1 ~ 0.7	12	25	0
0.1 ~ 0.8	-10	-27	0
0.2 ~ 0.9	-10	27	0

## 【相关主题】

· [CVI\\_ISP\\_GetStatisticsConfig](#)

## 40.2.2 CVI\_ISP\_GetStatisticsConfig

### 【描述】

获取 ISP 统计讯息配置。

### 【语法】

```
CVI_S32 CVI_ISP_GetStatisticsConfig(VI_PIPE ViPipe, ISP_STATISTICS_CFG_S *pstStatCfg);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstStatCfg	ISP 统计信息配置	输出

### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

### 【需求】

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

### 【注意】

无。

### 【举例】

无。

### 【相关主题】

- CVI\_ISP\_SetStatisticsConfig

## 40.2.3 CVI\_ISP\_GetAESTatistics

### 【描述】

获取 ISP AE 统计信息

### 【语法】

```
CVI_S32 CVI_ISP_GetAESTatistics(VI_PIPE ViPipe, ISP_AE_STATISTICS_S *pstAeStat);
```

### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstAeStat	AE 统计信息输出	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

```
//设定 AE window weight
#define AE_ZONE_ROW 15
#define AE_ZONE_COLUMN 17

ISP_AE_WIN_STATISTICS_CFG_S aeWinCfg;
CVI_U8 i,j;
VI_PIPE ViPipe = 0;
CVI_U8 u8Weighttable[AE_ZONE_ROW][AE_ZONE_COLUMN]={
{ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },
{ 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1 },
{ 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 1 },
{ 1, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 1 },
{ 1, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 1 },
{ 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1 },};

CVI_ISP_GetAEWinStatistics(ViPipe, &aeWinCfg);
for (i = 0; i < AE_ZONE_ROW; i++)
{
    for (j = 0; j < AE_ZONE_COLUMN; j++)
    {
        aeWinCfg.au8Weight[i][j] = u8Weighttable[i][j];
    }
}
CVI_ISP_SetAEWinStatistics(ViPipe, &aeWinCfg);
```

**【相关主题】**

无。

## 40.2.4 CVI\_ISP\_GetWBStatistics

**【描述】**

获取 ISP AWB 统计信息

**【语法】**

```
CVI_S32 CVI_ISP_GetWBStatistics(VI_PIPE ViPipe, ISP_WB_STATISTICS_S *pstAwbStat);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstAwbStat	AWB 统计信息	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

## 40.2.5 CVI\_ISP\_GetFocusStatistics

**【描述】**

获取 ISP AF 统计信息

**【语法】**

```
CVI_S32 CVI_ISP_GetFocusStatistics(VI_PIPE ViPipe, ISP_AF_STATISTICS_S *pstAfStat);
```

**【参数】**

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstAfStat	AF 统计信息	输出

**【返回值】**

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: cvi\_isp.h, cvi\_comm\_isp.h
- 库文件: libisp.so

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

## 40.3 数据类型

### 40.3.1 ISP\_STATISTICS\_CTRL\_U

**【说明】**

定义 ISP 统计值信息使能

**【定义】**

```
typedef union ISP_STATISTICS_CTRL_U {  
    CVI_U64 u64Key;  
    struct {  
        CVI_U64 bit1FEAeGloStat : 1; /* [0] */  
        CVI_U64 bit1FEAeLocStat : 1; /* [1] */  
        CVI_U64 bit1AwbStat1 : 1; /* [2] */  
        CVI_U64 bit1AwbStat2 : 1; /* [3] */  
        CVI_U64 bit1FEAfStat : 1; /* [4] */  
        CVI_U64 bit14Rsv : 59; /* [5:63] */  
    }  
} ISP_STATISTICS_CTRL_U;
```

## 【成员】

成员名称	描述
bit1FEAeGloStat	AE 全局统计值使能
bit1FEAeLocStat	AE 分块统计值使能
bit1AwbStat1	AWB 全局统计值使能
bit1AwbStat2	AWB 分块统计值使能
bit1FEAfStat	AF 分块统计值使能
bit14Rsv	保留位

## 【注意事项】

无。

## 【相关数据类型及接口】

无。

## 40.3.2 ISP\_AE\_STATISTICS\_CFG\_S

## 【说明】

AE 统计讯息配置

## 【定义】

```
typedef struct _ISP_AE_STATISTICS_CFG_S {  
    CVI_BOOL bHisStatisticsEnable; /*RW; Range:[0x0,0x1]*/  
    ISP_AE_CROP_S stCrop[AE_MAX_NUM];  
    ISP_AE_FACE_CROP_S stFaceCrop[FACE_WIN_NUM];  
    CVI_BOOL fast2A_ena; /*RW; Range:[0x0,0x1]*/  
    CVI_U8 fast2A_ae_low; /*RW; Range:[0x0,0xFF]*/  
    CVI_U8 fast2A_ae_high; /*RW; Range:[0x0,0xFF]*/  
    CVI_U16 fast2A_awb_top; /*RW; Range:[0x0,0xFFFF]*/  
    CVI_U16 fast2A_awb_bot; /*RW; Range:[0x0,0xFFFF]*/  
    CVI_U16 over_exp_thr; /*RW; Range:[0x0,0x3FF]*/  
    CVI_U8 au8Weight[AE_WEIGHT_ZONE_ROW][AE_WEIGHT_ZONE_COLUMN]; /*RW; Range:[0x0, 0xF]*/  
} ISP_AE_STATISTICS_CFG_S;
```

## 【成员】

成员名称	描述
enAESwitch	海思兼容参数, 目前不使用
stHistConfig	海思兼容参数, 目前不使用
enFourPlaneMode	海思兼容参数, 目前不使用
enHistMode	海思兼容参数, 目前不使用
enAverMode	海思兼容参数, 目前不使用
enMaxGainMode	海思兼容参数, 目前不使用
stCrop[AE_MAX_NUM];	AE 统计值输入图像裁减设定
au8Weight[AE_ZONE_ROW][AE_ZONE_COLUMN];	AE 统计值输入权重设定
bHisStatisticsEnable	AE/Hist sts 使能
u8StatisticsShiftBits	AE 统计值要缩小几个 bit
u16RGain	R 通道是否经过 RGain
u16GGain	G 通道是否经过 GGain
u16BGain	B 通道是否经过 BGain

#### 【相关数据类型及接口】

· CVI\_ISP\_SetAEWinStatistics

### 40.3.3 ISP\_AE\_CROP\_S

#### 【说明】

AE 统计值输入图像裁减设定

#### 【定义】

```
typedef struct _ISP_AE_CROP_S {
    CVI_BOOL bEnable;
    CVI_U16 u16X;
    CVI_U16 u16Y;
    CVI_U16 u16W;
    CVI_U16 u16H;
} ISP_AE_CROP_S;
```

#### 【成员】

成员名称	描述
bEnable	Crop 使能, 0 为不使能, 将设置整张图片作为裁减设定, 1 为使能
u16X	CropX 起始位置, 取值需大于 0
u16Y	CropY 起始位置, 取值需大于 0
u16W	Crop 宽度
u16H	Crop 高度

#### 【注意事项】

无

#### 【相关数据类型及接口】



无。

### 40.3.4 ISP\_AE\_FACE\_CROP\_S

#### 【说明】

AE face 统计值输入图像裁减设定

#### 【定义】

```
typedef struct _ISP_AE_FACE_CROP_S {  
    CVI_BOOL bEnable; /*RW; Range:[0x0,0x1]*/  
    CVI_U16 u16X; /*RW; Range:[0x00,0x1FFF]*/  
    CVI_U16 u16Y; /*RW; Range:[0x00,0x1FFF]*/  
    CVI_U8 u16W; /*RW; Range:[0x00,0xFF]*/  
    CVI_U8 u16H; /*RW; Range:[0x00,0xFF]*/  
} ISP_AE_FACE_CROP_S;
```

#### 【成员】

成员名称	描述
bEnable	Crop 使能，0 为不使能，将设置整张图片作为裁减设定，1 为使能
u16X	CropX 起始位置，取值需大于 0
u16Y	CropY 起始位置，取值需大于 0
u16W	Crop 宽度
u16H	Crop 高度

#### 【注意事项】

无

#### 【相关数据类型及接口】

无。

### 40.3.5 ISP\_WB\_STATISTICS\_CFG\_S

#### 【说明】

WB 统计讯息配置

#### 【定义】

```
typedef struct _ISP_WB_STATISTICS_CFG_S {  
    ISP_AWB_SWITCH_E enAWBSwitch;  
    CVI_U16 u16ZoneRow;  
    CVI_U16 u16ZoneCol;  
    CVI_U16 u16ZoneBin;  
    CVI_U16 au16HistBinThresh[4];  
    CVI_U16 u16WhiteLevel;
```

(下页继续)

(续上页)

```

CVI_U16 u16BlackLevel;
CVI_U16 u16CbMax;
CVI_U16 u16CbMin;
CVI_U16 u16CrMax;
CVI_U16 u16CrMin;
ISP_AWB_CROP_S stCrop;
} ISP_WB_STATISTICS_CFG_S;

```

**【成员】**

成员名称	描述
enAWBSwitch	ISP_AWB_AFTER_DG = 0 ISP_AWB_AFTER_Expander, ISP_AWB_AFTER_DRC, ISP_AWB_SWITCH_BUTT 设定 AWB block location
u16ZoneRow	AWB 统计窗口行数
u16ZoneCol	AWB 统计窗口列数
u16ZoneBin	海思兼容参数, 目前不使用
au16HistBinThresh	海思兼容参数, 目前不使用
u16WhiteLevel	AWB 亮点排除的 Threshold 设置
u16BlackLevel	AWB 暗点排除的 Threshold 设置
u16CbMax	不同 ISO 下 R/G 的最大值
u16CbMin	不同 ISO 下 R/G 的最小值
u16CrMax	不同 ISO 下 B/G 的最大值
u16CrMin	不同 ISO 下 B/G 的最小值

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.6 ISP\_AWB\_CROP\_S

**【说明】**

AWB 统计值输入图像裁减设定

**【定义】**

```

typedef struct _ISP_AWB_CROP_S {
    CVI_BOOL bEnable;
    CVI_U16 u16X;
    CVI_U16 u16Y;

```

(下页继续)

(续上页)

```

    CVI_U16 u16W;
    CVI_U16 u16H;
} ISP_AWB_CROP_S;

```

**【成员】**

成员名称	描述
bEnable	Crop 使能，假如 0 为不使能，将设置整张图片作为裁减设定，假如 1 为使能
u16X	CropX 起始位置，取值需大于 0 取值范围：[0, 图像宽度 - 16 * u16ZoneCol]
u16Y	CropY 起始位置，取值需大于 0 取值范围：[0, 图像高度 - 16 * u16ZoneRow]
u16W	Crop 宽度 取值范围：[16 * u16ZoneCol, 图像宽度]
u16H	Crop 高度 取值范围：[0, 图像高度 - 16 * u16ZoneRow]

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 40.3.7 ISP\_WB\_STATISTICS\_S

**【说明】**

定义 AWB 统计信息

**【定义】**

```

typedef struct _ISP_WB_STATISTICS_S {
    CVI_U16 u16GlobalR;
    CVI_U16 u16GlobalG;
    CVI_U16 u16GlobalB;
    CVI_U16 u16CountAll;
    CVI_U16 au16ZoneAvgR[AWB_ZONE_NUM];
    CVI_U16 au16ZoneAvgG[AWB_ZONE_NUM];
    CVI_U16 au16ZoneAvgB[AWB_ZONE_NUM];
    CVI_U16 au16ZoneCountAll[AWB_ZONE_NUM];
    ISP_AWB_GRID_INFO_S stGridInfo;
} ISP_WB_STATISTICS_S;

```

宏定义

```
#define AWB_ZONE_ORIG_ROW (32)
```

```
#define AWB_ZONE_ORIG_COLUMN (64)
```

```
#define AWB_ZONE_NUM (AWB_ZONE_ORIG_ROW * AWB_ZONE_ORIG_COLUMN)
```

**【成员】**

成员名称	描述
u16GlobalR	Bayer 域全局 R 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
u16GlobalG	Bayer 域全局 G 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
u16GlobalB	Bayer 域全局 B 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
u16CountAll	全局统计区域的像素个数 取值范围：[0, 0xffff] 数据类型：CVI_BOOL
au16ZoneAvgR	Bayer 域分区间 R 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
au16ZoneAvgG	Bayer 域分区间 G 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
au16ZoneAvgB	Bayer 域分区间 B 分量的平均值 取值范围：[0, 0x3ff] 数据类型：CVI_BOOL
au16ZoneCountAll	Bayer 域分区间的像素个数 取值范围：[0, 0xffff] 数据类型：CVI_BOOL
ISP_AWB_GRID_INFO_S	AWB 统计信息坐标信息

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.8 ISP\_AWB\_GRID\_INFO\_S

**【说明】**

AWB 统计信息坐标信息

**【定义】**

```
typedef struct ISP_AWB_GRID_INFO_S {
    CVI_U16 au16GridYPos[AWB_ZONE_ORIG_ROW + 1];
    CVI_U16 au16GridXPos[AWB_ZONE_ORIG_COLUMN + 1];
    CVI_U8 u8Status;
} ISP_AWB_GRID_INFO_S;
```

宏定义

(下页继续)

(续上页)

```
#define AWB_ZONE_ORIG_ROW (32)
#define AWB_ZONE_ORIG_COLUMN (32)
```

**【成员】**

成员名称	描述
au16GridYPos	暂未使用
au16GridXPos	暂未使用
u8Status	暂未使用

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.9 ISP\_FOCUS\_STATISTICS\_CFG\_S

**【说明】**

AF 统计讯息配置

**【定义】**

```
typedef struct _ISP_FOCUS_STATISTICS_CFG_S {
    ISP_AF_CFG_S stConfig;
    ISP_AF_H_PARAM_S stHParam_FIR0;
    ISP_AF_H_PARAM_S stHParam_FIR1;
    ISP_AF_V_PARAM_S stVParam_FIR;
} ISP_FOCUS_STATISTICS_CFG_S;
```

**【成员】**

成员名称	描述
stConfig	AF 全局配置参数
stHParam_FIR0	水平滤波器第一组 FIR 参数设置
stHParam_FIR1	水平滤波器第二组 FIR 参数设置
stVParam_FIR	垂直滤波器 FIR 参数设置

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 40.3.10 ISP\_AF\_CFG\_S

### 【说明】

定义 AF 统计值参数配置

### 【定义】

```
typedef struct _ISP_AF_CFG_S {
    CVI_BOOL bEnable;
    CVI_U16 u16Hwnd; /*RW; Range:[0x2, 0x11]*/
    CVI_U16 u16Vwnd; /*RW; Range:[0x2, 0xF]*/
    CVI_U8 u8HFltShift; /*RW; Range:[0x0, 0xF]*/
    CVI_S8 s8HFVltLpCoeff[FIR_H_GAIN_NUM]; /*RW; Range:[0x0, 0x1F]*/
    ISP_AF_RAW_CFG_S stRawCfg;
    ISP_AF_PRE_FILTER_CFG_S stPreFltCfg;
    ISP_AF_CROP_S stCrop;
    CVI_U8 u8HFltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 u8HFltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 u8VFltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U16 u16HighLumaTh; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 u8ThLow;
    CVI_U8 u8ThHigh;
    CVI_U8 u8GainLow; /*RW; Range:[0x0, 0xFE]*/
    CVI_U8 u8GainHigh; /*RW; Range:[0x0, 0xFE]*/
    CVI_U8 u8SlopLow; /*RW; Range:[0x0, 0xF]*/
    CVI_U8 u8SlopHigh; /*RW; Range:[0x0, 0xF]*/
} ISP_AF_CFG_S;
```

### 【成员】

成员名称	描述
bEnable	AF 使能, 0 为不使能, 1 为使能
u16Hwnd	AF 水平方向窗口数, 最大可设置为 17 取值范围: [0x2, 0x11] 数据类型: CVI_U16
u16Vwnd	AF 垂直方向窗口数, 最大可设置为 15 取值范围: [0x2, 0xf] 数据类型: CVI_U16
u8HFltShift	AF 低通滤波器统计值移位寄存器值 取值范围: [0, 16] 数据类型: CVI_U8
s8HFVltLpCoeff[FIR_H_GAIN_NUM]	AF 低通滤波器系数, 用于控制 FIR 滤波器的频率响应 取值范围: [-32, 31] 数据类型: CVI_S8
stRawCfg	AF Bayer 域相关配置
stPreFltCfg	AF 预滤波处理使能设定
stCrop	AF 输入图像的裁剪配置
u16HighLumaTh	AF 高亮点统计值 Threshold 设置 取值范围: [0, 4095]

### 【注意事项】

无。

**【相关数据类型及接口】**

无。

### 40.3.11 ISP\_AF\_RAW\_CFG\_S

**【说明】**

AF Bayer 域影像预处理配置

**【定义】**

```
typedef struct _ISP_AF_RAW_CFG_S {  
    CVI_U8 PreGammaEn;  
    CVI_U8 PreGammaTable[AF_GAMMA_NUM];  
} ISP_AF_RAW_CFG_S;
```

**【成员】**

成员名称	描述
PreGammaEn	AF 模块 Gamma 使能, 0 为不使能, 1 为使能
PreGammaTable[AF_GAMMA_NUM]	AF 模块 Gamma table 设置

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.12 ISP\_AF\_PRE\_FILTER\_CFG\_S

**【说明】**

AF Bayer 域影像滤波预处理配置

**【定义】**

```
typedef struct _ISP_AF_PRE_FILTER_CFG_S {  
    CVI_BOOL PreFltEn;  
} ISP_AF_PRE_FILTER_CFG_S;
```

**【成员】**

成员名称	描述
PreFltEn	AF 模块滤波预处理使能, 0 为不使能, 1 为使能

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.13 ISP\_AF\_CROP\_S

**【说明】**

AF 统计值输入图像裁减设定

**【定义】**

```
typedef struct _ISP_AF_CROP_S {  
    CVI_BOOL bEnable;  
    CVI_U16 u16X;  
    CVI_U16 u16Y;  
    CVI_U16 u16W;  
    CVI_U16 u16H;  
} ISP_AF_CROP_S;
```

**【成员】**

成员名称	描述
bEnable	Crop 使能，0 为不使能，将设置整张图片作为裁减设定，1 为使能
u16X	CropX 起始位置 取值范围：[8, 图像宽度 - 8]
u16Y	CropY 起始位置 取值范围：[2, 图像高度 - 2]
u16W	Crop 宽度 取值范围：[16 * u16ZoneCol, 图像宽度 - 16]
u16H	Crop 高度 取值范围：[16 * u16ZoneRow, 图像高度 - 4]

**【注意事项】**

无。

**【相关数据类型及接口】**

无。



### 40.3.14 ISP\_AF\_H\_PARAM\_S

**【说明】**

定义 AF 水平滤波器参数设置

**【定义】**

```
typedef struct _ISP_AF_H_PARAM_S {  
    CVI_S8 s8HFltHpCoeff[FIR_H_GAIN_NUM];  
} ISP_AF_H_PARAM_S;
```

**【成员】**

成员名称	描述
s8HFltHpCoeff[FIR_H_GAIN_NUM]	AF 水平滤波器系数，用于控制 FIR 滤波器的频率响应 取值范围: [-32, 31]

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.15 ISP\_AF\_V\_PARAM\_S

**【说明】**

定义 AF 垂直滤波器参数设置

**【定义】**

```
typedef struct _ISP_AF_V_PARAM_S {  
    CVI_S8 s8VFltHpCoeff[FIR_V_GAIN_NUM];  
} ISP_AF_V_PARAM_S;
```

**【成员】**

成员名称	描述
s8VFltHpCoeff[FIR_V_GAIN_NUM]	AF 垂直滤波器系数，用于控制 FIR 滤波器的频率响应 取值范围: [-32, 31]

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.16 ISP\_STATISTICS\_CFG\_S

**【说明】**

ISP 统计讯息配置

**【定义】**

```
typedef struct _ISP_STATISTICS_CFG_S {  
    ISP_STATISTICS_CTRL_U unKey;  
    ISP_AE_STATISTICS_CFG_S stAECfg;  
    ISP_WB_STATISTICS_CFG_S stWBCfg;  
    ISP_FOCUS_STATISTICS_CFG_S stFocusCfg;  
} ISP_STATISTICS_CFG_S;
```

**【成员】**

成员名称	描述
unKey	统计信息使能
stAECfg	AE 统计信息配置
stWBCfg	WB 统计信息配置
stFocusCfg	AF 统计信息配置

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.17 ISP\_FOCUS\_ZONE\_S

**【说明】**

定义 AF 计算出的统计结果

**【定义】**

```
typedef struct _ISP_FOCUS_ZONE_S {  
    CVI_U16 u16HlCnt;  
    CVI_U64 u64h0;  
    CVI_U64 u64h1;  
    CVI_U32 u32v0;  
} ISP_FOCUS_ZONE_S;
```

**【成员】**

成员名称	描述
u16HiCnt	AF 分区间内统计超过高亮点阈值的点数值，将设置整张图片作为裁减设定，1 为使能
u64h0	AF 分区间内统计水平方向第一组 FIR 滤波器的结果
u64h1	AF 分区间内统计水平方向第二组 FIR 滤波器的结果
u32v0	AF 分区间内统计垂直方向 FIR 滤波器的结果

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.18 ISP\_FE\_FOCUS\_STATISTICS\_S

**【说明】**

定义 AF FE 提供的统计讯息

**【定义】**

```
typedef struct _ISP_FE_FOCUS_STATISTICS_S {  
    ISP_FOCUS_ZONE_S stZoneMetrics[AF_ZONE_ROW][AF_ZONE_COLUMN];  
} ISP_FE_FOCUS_STATISTICS_S;
```

**【成员】**

成员名称	描述
stZoneMetrics	ISP AF 的分块统计信息

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 40.3.19 ISP\_AF\_STATISTICS\_S

**【说明】**

定义 AF 提供的所有统计信息

**【定义】**

```
typedef struct _ISP_AF_STATISTICS_S {  
    ISP_FE_FOCUS_STATISTICS_S stFEAFStat;  
} ISP_AF_STATISTICS_S;
```

**【成员】**

成员名称	描述
stFEAFStat	AF 在 Bayer 域的统计信息

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

# 41 查询内部状态消息

## 41.1 概述

本章节说明 Inner State information 的相关接口。此接口的作用为提供用户查询系统内部状态以及数个与 ISO 相关的参数目前所设置的真实数值。

使用者可在调试过程中透过此接口获取 ISO 相关参数目前填入的真实数值以确认参数是否正确配置。透过此接口只能获得数值，无法改变相关参数。

## 41.2 API 参考

- `CVI_ISP_QueryInnerStateInfo`: 获取系统内部信息以及 ISO 相关参数目前设定的真实数值。

### 41.2.1 CVI\_ISP\_QueryInnerStateInfo

#### 【描述】

获取系统内部信息以及 ISO 相关参数目前设定的真实数值

#### 【语法】

```
CVI_S32 CVI_ISP_QueryInnerStateInfo(VI_PIPE ViPipe, ISP_INNER_STATE_INFO_S_
↪ *pstInnerStateInfo);
```

#### 【参数】

参数名称	描述	输入/输出
ViPipe	VI_PIPE 号	输入
pstInnerStateInfo	内部信息与目前实际参数设定值	输出

#### 【返回值】

返回值	描述
0	成功。
非 0	失败，其值为错误码。

**【需求】**

- 头文件: `cvi_isp.h`, `cvi_comm_isp.h`
- 库文件: `libisp.so`

**【注意】**

无。

**【举例】**

无。

**【相关主题】**

无。

## 41.3 数据类型

- `ISP_INNER_STATE_INFO_S`: 定义内部信息与目前实际参数设定值

### 41.3.1 `ISP_INNER_STATE_INFO_S`

**【说明】**

定义内部信息与目前实际参数设定值

**【定义】**

```
#define MAX_HIST_BINS 256
#define MAX_EXPOSURE_RATIO 256
#define LTM_DARK_CURVE_NODE_NUM 257
#define LTM_BRIGHT_CURVE_NODE_NUM 513
typedef struct _ISP_INNER_STATE_INFO_S {
    CVI_U32 blcOffsetR;
    CVI_U32 blcOffsetGr;
    CVI_U32 blcOffsetGb;
    CVI_U32 blcOffsetB;
    CVI_U32 blcGainR;
    CVI_U32 blcGainGr;
    CVI_U32 blcGainGb;
    CVI_U32 blcGainB;
    CVI_S32 ccm[9];
    CVI_U16 drcGlobalToneBinNum;
    CVI_U16 drcGlobalToneBinSEStep;
    CVI_U32 drcGlobalTone[LTM_GLOBAL_CURVE_NODE_NUM];
    CVI_U32 drcDarkTone[LTM_DARK_CURVE_NODE_NUM];
    CVI_U32 drcBrightTone[LTM_BRIGHT_CURVE_NODE_NUM];
    CVI_BOOL bWDRSwitchFinish;
    // For 2TO1/3TO1/4TO1 use.
    CVI_U32 u32WDRExpRatioActual[ISP_WDR_FRAME_IDX_SIZE];
    ISP_MESH_SHADING_GAIN_LUT_S mlscGainTable;
} ISP_INNER_STATE_INFO_S;
```

## 【成员】

成员名称	描述
wdrHistBinNum	目前画面 wdr 分布直方图的数组数量
blcOffsetR	目前黑电平在 R 分量扣除的值
blcOffsetGr	目前黑电平在 Gr 分量扣除的值
blcOffsetGb	目前黑电平在 Gb 分量扣除的值
blcOffsetB	目前黑电平在 B 分量扣除的值
blcGainR	目前 ISP 在 R 分量的数字增益
blcGainGr	目前 ISP 在 Gr 分量的数字增益
blcGainGb	目前 ISP 在 Gb 分量的数字增益
blcGainB	目前 ISP 在 B 分量的数字增益
Ccm[9]	目前 ISP 使用的色彩还原矩阵真实数值
wdrHistogramBefore[MAX_HIST_BINS * MAX_EXPOSURE_RATIO]	目前 wdr 的分布直方图
wdrHistogramAfter[MAX_HIST_BINS * MAX_EXPOSURE_RATIO]	目前 wdr 分布直方图所产生的色调映像曲线
drcDarkTone[LTM_DARK_CURVE_NODE_NUM]	目前 ISP 使用的暗区色调映像曲线
drcBrightTone[LTM_BRIGHT_CURVE_NODE_NUM]	目前 ISP 使用的亮区色调映像曲线
bWDRSwitchFinish	标示 sensor mode 切换是否完成

## 【注意事项】

无。

## 【相关数据类型及接口】

无。

# 42 Debug

---



# 43 错误码

## 43.1 Proc 调试信息说明

## 43.2 概述

调试信息采用 procfs，可以根据上层设定的 proc param 和 proc level，实时反映 ISP 模组内的设定参数，3A 统计值，3A 运算结果等信息，供开发人员定位问题和分析问题使用。

## 43.3 使用方法

### 【文件目录】

/proc/cvitek/isp

### 【开启方法】

在呼叫 CVI\_ISP\_MemInit 后，呼叫 CVI\_ISP\_SetCtrlParam 设定 ISP 控制参数 u32ProcParam=n，其中 n 不能为 0，n 表示收集 ISP 信息的频率，每隔 n 帧收集一次 ISP 信息，默认值为 30

呼叫 CVI\_ISP\_SetCtrlParam 设定 ISP 控制参数 u32ProcLevel=m，其中 m 的范围为 **【0, 3】** m=0 时表示 proc 功能关闭，m=1 时表示打印 level1 级别的资讯，m=2 时表示打印 level2 级别的资讯，m=3 时表示打印 level3 级别的资讯，level 级别越高，表示打印的资讯越详细，而且高级别的打印内容一定包含低级别的打印内容，默认值为 0

在使用 ISP\_TOOL\_DAEMON 时，若要开启 proc 功能，可以在跑 ISP\_TOOL\_DAEMON 前 export PROC\_LEVEL=n(n 为 1-3,n=0 不开启)

### 【查看信息方法】

ISP module 正常运行后

在控制台通过调用 cat /proc/cvitek/isp，相关 log 随后会打印于控制台，cp /proc/cvitek/isp 到 pc 端查看。

## 43.4 ISP

### 43.4.1 LEVEL1 级别调试信息分析

ISP 的各模块参数打印

【参数说明】

参数		描述
MOD- ULE/CTRLPARAM	ProcParam	表示收集 ISP 信息的收集频率
	ProcLevel	表示 ISP proc 的 level 级别
	AESatIntvl	表示 ISP AE 统计信息更新频率
	AWBStatIntvl	表示 ISP AWB 统计信息更新频率
	AFStatIntvl	表示 ISP AF 统计信息更新频率
	UpdatePos	表示中断更新位置是用帧起始中断还是在帧结束中断，0 是帧起始
	IntTimeOut	表示获取 ISP 中断超时的最大时间
	PwmNumber	表示 pwm 使用情况
	PortIntDelay	表示 Port 中断延时时间
FSWDR	Enable	WDR 模块使能
	MotionCom- pEnable	WDR 运动侦测使能开关
	CombineS- NRAwareEn	具 SNR 感知的宽动态融合模式使能，能根据短帧噪声自适应取决长短帧融合比例
	CombineS- NRAwareLowThr	短帧噪声 SNR 自适应低临界值。当短帧估测噪声低于低临界值则进行一般长短帧融合，噪声介于低临界值与高临界值间则按比例进行 SNR 自适应融合。
	CombineS- NRAwareHigh- Thr	当短帧估测噪声高于高临界值则根据 SNRAwareToleranceLevel 强度进行 SNR 自适应融合，噪声介于低临界值与高临界值间则按比例进行 SNR 自适应融合。
	CombineS- NRAwareSm- Level	短帧噪声 SNR 自适应之时间域变化平滑程度
	ExposureRatio	WDR 长短曝的曝光比
	ShortMaxVal	WDR 宽动态的最大值
	isManualMode	手动模式或自动模式
	Combine- LongThr	长曝光临界值，低于该临界值的图像数据将只选择长曝光数据合成 WDR 影像
	CombineShort- Thr	短曝光临界值，超过该临界值的图像数据将只选择短曝光数据合成 WDR 影像
	CombineM- inWeight	长短曝图像数据融合最低权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多
	Combine- MaxWeight	长短曝图像数据融合最高权重值。权重值越大，融合时长曝光占的比重越多，反之则短曝光占的比重越多

下页继续

表 43.1 – 续上页

参数		描述
	MergeMode	WDR 运动侦测模式 0: 运动侦测资讯取长短帧运动资讯最大值 1: 运动侦测资讯为长短帧运动资讯之等比例融合, 比例由 MergeModeAlpha 定义
	MergeModeAlpha	MergeMode 模式为 1 的移动资讯融合比例
	CombineS-NRAware-TolLevel	短帧噪声 SNR 自适应之噪声容忍强度
Shading	MeshEnable	MeshLSC 功能使能
	MeshisManualMode	手动模式或自动模式
	MeshStr	LSC 补偿强度
	MeshLscGain-Lut Size	色温自适应 LSC 补偿增益表数量
	MeshLscGain-LutColorTemp	色温自适应 LSC 补偿增益表所对应之色温
	RadialEnable	RadialLSC 功能使能
	RadialCenterX	图像传感器镜心 X 方向坐标
	RadialCenterY	图像传感器镜心 Y 方向坐标
	RadialStr	LSC 补偿强度
	Low-RadialGainLut	LSC R adius 形式补偿增益表
	High-RadialGainLut	LSC R adius 形式补偿增益表
DCI	Enable	DCI 模块使能
	Speed	Smooth 强度, 值越高, 则变化越慢
	DciStrength	用来控制 DCI 的强度, 值越大, 对比度越大。
	isManualMode	手动模式或自动模式
	ContrastGain	保留每个 BIN 最低斜率, 值越大斜率越小
	BlcThr	用来决定暗区范围的阈值。值越大, 包含的暗区范围越大。
	WhtThr	用来决定亮区范围的阈值。值越小, 包含的亮区范围越大。
	BlcCtrl	用来决定暗区的对比度。数值为 256 时, 暗区对比度不变。比 256 大时, 值越大, 暗区对比度越大; 反之, 比 256 小时, 值越小, 暗区对比度越小。
	WhtCtrl	用来决定亮区的对比度。数值为 256 时, 亮区对比度不变。比 256 大时, 值越大, 亮区对比度越大; 反之, 比 256 小时, 值越小, 亮区对比度越小。
Dehaze	Enable	Dehaze 功能使能
	isManualMode	手动模式或自动模式
	Strength	用来控制 Dehaze 的强度。值越大, 去雾强度越强
BlackLevel	Enable	BLC 模块使能
	isManualMode	手动模式或自动模式
	PreOffsetR	BLC R 像素暗电流值
	PreOffsetGr	BLC GR 像素暗电流值

下页继续

表 43.1 – 续上页

参数		描述
	PreOffsetGb	BLC Gb 像素暗电流值
	PreOffsetB	BLC B 像素暗电流值
	PreGainR	BLC R 像素暗电流补偿增益
	PreGainGr	BLC Gr 像素暗电流补偿增益
	PreGainGb	BLC Gb 像素暗电流补偿增益
	PreGainB	BLC B 像素暗电流补偿增益
	PostOffsetR	BLC R 像素暗电流值
	PostOffsetGr	BLC Gr 像素暗电流值
	PostOffsetGb	BLC Gb 像素暗电流值
	PostOffsetB	BLC B 像素暗电流值
	PostGainR	BLC R 像素暗电流补偿增益
	PostGainGr	BLC Gr 像素暗电流补偿增益
	PostGainGb	BLC Gb 像素暗电流补偿增益
	PostGainB	BLC B 像素暗电流补偿增益
DPC	Enable	DPC 模块使能
	isManualMode	手动模式或自动模式
	ClusterSize	群聚坏点面积上限，值越高越能修正群聚坏点，但可能会造成高频区域解像力的衰减
	Bright-DefToNor-PixRatio	可视亮坏点值与周围像素的倍率
	DarkDefToNor-PixRatio	可视暗坏点值与周围像素的倍率
	FlatThreR	R 通道判别平坦区临界值，值越小越能保留边缘信息
	FlatThreG	B 通道判别平坦区临界值，值越小越能保留边缘信息
	FlatThreB	B 通道判别平坦区临界值，值越小越能保留边缘信息
	FlatThreMinG	G 通道判别平坦区最小临界值
	FlatThreMinRB	RB 通道判别平坦区最小临界值
TNR	Enable	TNR 模块使能
	DeflickerMode	抗闪烁模式
	DeflickerToleranceLevel	抗闪烁模式 mode4 抗闪容忍值，数值越大越能抗闪烁，但亮区的移动残影也会增加。
	LowMtPrtEn	空域降噪微小运动保护使能
	isManualMode	手动模式或自动模式
	TnrStrength0	长曝光 TNR 强度增益
	MapThdLow0	长曝光 TNR 强度上限
	MapThdHigh0	长曝光 TNR 强度下限
	Brightness-NoiseLevelLE	长曝光亮度噪声容忍值
	Brightness-NoiseLevelSE	短曝光亮度噪声容忍值
	RNoiseLevel0	长曝光红色通道噪声容忍值
	RNoiseHiLevel0	长曝光红色通道亮部噪声容忍值
	GNoiseLevel0	长曝光绿色通道噪声容忍值
	GNoiseHiLevel0	长曝光绿色通道亮部噪声容忍值
	BNoiseLevel0	长曝光蓝色通道噪声容忍值

下页继续

表 43.1 – 续上页

参数		描述
	BNoiseHiLevel0	长曝光蓝色通道亮部噪声容忍值
	RNoiseLevel1	短曝光红色通道噪声容忍值
	RNoiseHiLevel1	短曝光红色通道亮部噪声容忍值
	GNoiseLevel1	短曝光绿色通道噪声容忍值
	GNoiseHiLevel1	短曝光绿色通道亮部噪声容忍值
	BNoiseLevel1	短曝光蓝色通道噪声容忍值
	BNoiseHiLevel1	短曝光蓝色通道亮部噪声容忍值
	LowMtPrtLevel	保护限值
	L2mIn0	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。
	L2mOut0	长曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。
	L2mIn1	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义灰度等级，值越大灰度越高。
	L2mOut1	短曝光 TNR 亮度对强度增益特性表。由四组数值组成的数组。定义强度增益，值越大强度越强。
	PrtctIn0	长曝光 TNR 动量对拖尾消除程度特性表，由四组数值组成的数组。定义运动等级对运动物体拖尾的消除程度，值越大运动幅度越强。
	PrtctOut0	长曝光 TNR 动量对拖尾消除程度特性表，由四组数值组成的数组。定义运动等级对运动物体拖尾的消除程度，值越小拖尾消除程度越强。
	PrtctIn1	短曝光 TNR 动量对拖尾消除程度特性表，由四组数值组成的数组。定义运动等级对运动物体拖尾的消除程度，值越大运动幅度越强。
	PrtctOut1	短曝光 TNR 动量对拖尾消除程度特性表，由四组数值组成的数组。定义运动等级对运动物体拖尾的消除程度，值越小拖尾消除程度越强。
	LowMtPrtIn	定义运动等级
	LowMtPrtOut	值越大，拖影越不明显，噪声越明显
CAC	Enable	CAC 模块使能
	VarThr	边缘侦测的阈值。值越小，越多区域被判断为边缘。
	PurpleDetRange	紫边侦测的阈值。值越大，越多区域被判断为紫边。
	PurpleCb	紫色在 Cb domain 的坐标。
	PurpleCr	紫色在 Cr domain 的坐标。
	GreenCb	绿色在 Cb domain 的坐标。
	GreenCr	绿色在 Cr domain 的坐标。
	isManualMode	手动模式或自动模式
	DePurpleStr	紫边侦测的阈值。值越大，越多区域被判断为紫边。
CNR	Enable	CNR 模块使能。
	isManualMode	手动模式或自动模式
	CnrStr	色噪去噪强度。值越大，色噪去噪强度越大。
	NoiseSuppressStr	色噪抑制强度。值越大，色噪去除强度越大。
	NoiseSuppressGain	色噪抑制强度增益。值越小，色噪去噪强度越大。

下页继续

表 43.1 – 续上页

参数		描述
	FilterType	色噪去噪滤波器强度。值越大，色噪去除强度越大。
	MotionNrStr	调节物体运动区域的色噪去噪强度。值越大，运动区域的色噪越少。
	DetailSmooth-Mode	去噪细节平滑功能使能
Sharpen	Enable	Y Sharpen 模块使能
	EdgeGain	边缘的增强参数值越大时，边缘锐化强度越大
	TextureGain	细节纹理的增强参数值越大时，细节纹理锐化强度越大
	EdgeThr	边缘和噪声的分界阈值，大于该值被视为是边缘，小于该值则被视为噪声该值越大，越少的边缘被增加；该值越小，越多的边缘被增强。曝光增益越大时，该值建议设置越大
	TextureThr	细节纹理和噪声的分界阈值，大于该值被视为是细节纹理，小于该值则被视为噪声
	LumaAdpCoringEn	自动亮度噪声抑止阈值开关
	LumaAdpGainEn	亮度锐化权重使能
	DeltaAdpGainEn	锐化权重使能
	WdrCoringCompensationEn	亮度锐化噪声值在 wdr 模式的补偿使能。
	WdrCoringCompensation-Mode	亮度锐化噪声值在 wdr 模式的补偿模式。0: 根据 WdrCoringHighThrd 与 WdrCoringLowthd 补偿亮度锐化噪声。1: 根据 DRC tone mapping curve 自动补偿亮度锐化噪声。
	WdrCoringToleranceLevel	亮度锐化噪声值在 wdr 模式补偿的容忍值。数值越小锐化的效果越明显但也越容易将噪声锐化。数值越大则不易将噪声锐化，但锐化的效果则越不明显。
	WdrCoringHighThr	亮度锐化噪声 wdr 模式补偿的高临界值。亮度低于此临界值则不进行噪声补偿。亮度高于此临界值则根据 WdrCoringToleranceLevel 数值进行噪声补偿。亮度介于 WdrCoringLowThrd 与 WdrCoringHighThrd 之间则按比例进行补偿。
	WdrCoringLowThr	亮度锐化噪声 wdr 模式补偿的低临界值。亮度低于此临界值则不进行噪声补偿。亮度高于此临界值则根据 WdrCoringToleranceLevel 数值进行噪声补偿。亮度介于 WdrCoringLowThrd 与 WdrCoringHighThrd 之间则按比例进行补偿。
	isManualMode	手动模式或自动模式
	EdgeFreq	图像具方向性的边缘频段控制
	TextureFreq	图像无方向性的细节纹理频段控制
	GlobalGain	全局锐化权重
	OverShootThr	白边锐化上限幅度
	UnderShootThr	黑边锐化下限幅度

下页继续



表 43.1 – 续上页

参数		描述
	YNoiseLevel	亮度锐化噪声值放大倍率，一倍为 64。数值越大亮度锐化噪声越被放大，反之则缩小。
	DeltaAdpGain	锐化权重
	LumaAdpCor-ing	亮度锐化噪声值，细节纹理或边缘的增强会排除该容忍值所贡献的增强。数值越小锐化的效果越明显但也越容易将噪声锐化。数值越大则不易将噪声锐化，但锐化的效果则越不明显。
	LumaAdpGain	亮度锐化权重
Saturation	isManualMode	手动模式或自动模式
	Saturation	饱和度
Gamma	Enable	Gamma 功能使能
	enCurveType	Gamma 曲线类型
	isManualMode	手动模式或自动模式
HSV	Enable	Saturation Tuning 使能
	isManualMode	手动模式或自动模式
	SatCoringLin-earTh	饱和度的 coring 值
	SatCoringLin-earLmt	最大输出饱和度
CCM	Enable	CCM 模块使能
	isManualMode	手动模式或自动模式
	ISOActEnable	低照度下 CCM Bypass 功能使能
	TempActEnable	高低色温下 CCMBypass 功能使能。
	CCMTabNum	当前配置的 CCM 矩阵个数
	CCM	颜色校正矩阵
	CCMTab	不同色温下的 CCM 矩阵系数
YNR	Enable	YNR 模块使能
	CoringPara-mEnable	控制是否使用手动 coring
	isManualMode	手动模式或自动模式
	WindowType	去噪滤波局域程度。其值越小，作用越局域
	DetailSmooth-Mode	去噪细节平滑功能使能
	NoiseSup-pressStr	噪声抑制强度。值越大，亮噪去除强度越大
	FilterType	去噪滤波器强度。值越大，亮噪去除强度越大
	MotionThr	物体移动量阈值。值越小，侦测为运动的区域范围越大。
	MotionNrPos-Gain	调节在大于移动量阈值的区域之去噪强度。值越大，噪声保留越少。
	MotionNrNeg-Gain	调节在小于移动量阈值的区域之去噪强度。值越小，噪声保留越少。
	VarThr	侦测边缘的阈值。值越大，判断为边缘的数量越少。
	CoringWgtLF	调节在低频区域的随机噪声强度。值越大，在低频区域保留的噪声越多。
	CoringWgtHF	调节在高频区域的随机噪声强度。值越大，在高频区域的保留的噪声越多。

下页继续

表 43.1 – 续上页

参数		描述
	NonDirFiltStr	调节在低频区的去噪强度。值越大，在低频区域去除的噪声越多。
	VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大，在水平和垂直边缘去除的噪声越多。
	AaDirFiltStr	调节在对角线边缘的去噪强度。值越大，在对角线边缘去除的噪声越多。
	NoiseCoring-BaseLuma	运动区亮度值。
	NoiseCoring-BaseOffset	运动区亮度噪声容忍值，运动区的判断与 TNr 运动区侦测连动。值越大，对运动区去噪强度越大。
	NoiseCoringAdvLuma	静止区亮度值
	NoiseCoringAdvOffset	静止区亮度噪声容忍值，静止区的判断与 TNr 运动区侦测连动。值越大，对静止区去噪强度越大。
WDRExposure-Attr	ExpRatioType	仅在多帧合成 WDR 模式下有效。OP_TYPE_AUTO: 根据场景自动计算长短帧曝光比; OP_TYPE_MANUAL: 手动配置长短帧曝光比。
	ExpRatio	仅在多帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时, au32ExpRatio 无效。当 enExpRatioType 为 OP_TYPE_MANUAL 时, au32ExpRatio 为可擦写, 表示多帧合成 WDR 相邻 2 帧曝光比期望值。取值范围: <b>【0x40,0xFFF】</b>
	ExpRatioMax	仅在多帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMax 表示最长帧与最短帧曝光时间比值的最大值。当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMax 无效。6bit 小数精度, 0x40 表示曝光比为 1 倍。
	ExpRatioMin	仅在多帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时, u32ExpRatioMin 表示长帧曝光时间与短帧曝光时间比值的最小值。当 enExpRatioType 为 OP_TYPE_MANUAL 时, u32ExpRatioMin 无效。格式为无符号 6.6bit 定点, 0x40 表示长帧曝光时间与短帧曝光时间的比值为 1 倍。默认值为 0x40。取值范围: <b>【0x40, u32ExpRatioMax】</b>
	Tolerance	曝光比容忍值, 仅在两帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 表示场景动态范围变化在一定范围内时, 曝光比保持不变。默认值为 0xC。取值范围: <b>【0x0, 0xFF】</b>
	Speed	自动曝光比调节速度, 仅在两帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时, 该值越大, 自动曝光比调节速度越快。默认值为 0x20。取值范围: <b>【0x0, 0xFF】</b>

下页继续



表 43.1 – 续上页

参数		描述
	RatioBias	曝光比偏差值，仅在多帧合成 WDR 模式下有效。当 enExpRatioType 为 OP_TYPE_AUTO 时，该值越大，自动曝光比越大。默认值为 0x400，表示不对自动曝光比算法的计算结果进行调整。经过该值调整的曝光比会受到曝光比最大/最小值的限制。取值范围：【0x0, 0xFFFF】
	SECompensation	调整短帧画面的目标亮度值取值范围：【0x0, 0xFF】
	SEHisThr	计算短帧 frame 超过长帧 frame 的曝光比阈值 (1x =64)
	SEHis255CntTargetDownRatio	短帧 histogram bin 255 数量大于此阈值，则调降短帧的目标亮度
	SEHis255CntTargetUpRatio	短帧 histogram bin 255 数量小于此阈值，则将调降的短帧的目标亮度恢复原来的目标亮度
	LEHisCntTargetDownRatio	长帧 histogram bin 255 数量大于此阈值，则调降的长帧的目标亮度
	LEHisCntTargetUpRatio	长帧 histogram bin 255 数量小于此阈值，则将调降的长帧的目标亮度恢复原来的目标亮度
	LEAdjustTargetMin	长帧各 LV 的 target min 40,40, 40, 40, 40, 40, 45, 50, 60, 60, 60, 60, 60, 60, 60
	LEAdjustTargetMax	长帧各 LV 的 target max 50,50, 50, 50, 50, 50, 60, 70, 85, 100, 110, 110, 120, 120, 120, 120
	SEAdjustTargetMin	短帧各 LV 的 target min 20,20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20
	SEAdjustTargetMax	短帧各 LV 的 target max 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60
	DiffPixelNum	长帧 histogram bin 255 的 pixel 数阈值，超过此值，则降低长曝的目标亮度
	LELowBinThr	长帧亮度大于此阈值的 window 才能加入 AE 长帧的测光
	LEHighBinThr	长帧亮度小于此阈值的 window 才能加入 AE 长帧的测光
	SELowBinThr	短帧亮度大于此阈值的 window 才能加入 AE 短帧的测光
	SEHighBinTh	短帧亮度小于此阈值的 window 才能加入 AE 短帧的测光
ExposureAttr	ByPass	AE 模块 bypass 功能使能，默认为 CVI_FALSE FSWDR 模式下，表示当前最短帧 (VS) 曝光时间。
	OpType	自动曝光或手动曝光开关，默认为 OP_TYPE_AUTO。
	AERunInterval	AE 算法运行的间隔取值范围为【1,255】。取值为 1 时表示每帧都运行 AE 算法；取值为 2 时表示每 2 帧运行 1 次 AE 算法，依此类推。建议该值设置不要大于 2，否则 AE 调节速度会受到影响。WDR 模式时，该值建议设置为 1，这样 AE 收敛会更加平滑。该值默认为 1。线性模式不用关注该值。
	ExpTimeOp-Type	手动曝光时间使能，默认值为 OP_TYPE_AUTO

下页继续

表 43.1 – 续上页

参数		描述
	AGainOpType	手动 sensor 模拟增益使能, 默认值为 OP_TYPE_AUTO
	DGainOpType	手动 sensor 数字增益使能, 默认值为 OP_TYPE_AUTO
	ISPDGainOp-Type	手动 ISP 数字增益, 10bit 小数精度, 默认值为 0x400。
	ExpTime	手动曝光时间, 以微秒 (us) 为单位, 默认值为 0x4000
	AGain	手动 sensor 模拟增益, 10bit 小数精度, 默认值为 0x400。
	DGain	手动 sensor 数字增益, 10bit 小数精度, 默认值为 0x400。
	ISPDGain	手动 ISP 数字增益, 10bit 小数精度, 默认值为 0x400。
	ISONumOp-Type	ISO num 手动/自动模式切换
	UseISONum	手动模式时, 使用 ISO num 的型式来控制增益
	ISONum	手动模式时, 设定的 ISO num
	ExpTimeRange-Max	曝光时间范围, 设置最大值, 以微秒 (us) 为单位。
	ExpTimeRangeMin	曝光时间范围, 设置最小值, 以微秒 (us) 为单位。
	AGainRange-Max	Sensor 模拟增益范围, 设置最大值, 10bit 小数精度。
	AGainRangeMin	Sensor 模拟增益范围, 设置最小值, 10bit 小数精度。
	DGainRange-Max	Sensor 数字增益范围, 设置最大值, 10bit 小数精度。
	DGainRangeMin	Sensor 数字增益范围, 设置最小值, 10bit 小数精度。
	ISPDGainRangeMax	ISP 数字增益范围, 设置最大值, 10bit 小数精度。
	ISPDGainRangeMin	ISP 数字增益范围, 设置最小值, 10bit 小数精度。
	SysGainRangeMin	系统增益范围, 设置最小值, 10bit 小数精度
	SysGainRangeMax	系统增益范围, 设置最大值, 10bit 小数精度
	Speed	自动曝光调整时的速度。
	BlackSpeedBias	画面由暗到亮 AE 调节速度的偏差值, 该值越大, 画面从暗到亮的速度越快。
	Tolerance	自动曝光调整时对画面亮度的容忍偏差。
	Compensation	自动曝光调整时的目标亮度。
	EVBias	自动曝光调整时的曝光量偏差值, 10bit 小数精度
	AEStrategy-Mode	自动曝光策略, 高光优先或低光优先。
	AntiflickerEnable	抗闪属性设置。默认抗闪不使能。
	AntiflickerFrequency	抗闪频率值

下页继续

表 43.1 – 续上页

参数		描述
	DebugMode	AE 的 debug mode 参数
	IRCutOnLv	不使用 IR Cut 的环境 Lv 值, 精度为 100
	IRCutOffLv	使用 IR Cut 的环境 Lv 值, 精度为 100
	HistRatioSlope	目前 frame 的亮度与高光区亮度的混合权重
	MaxHistOffset	目前 frame 的亮度与高光区亮度的混合后, 提升的亮度范围
	ISONum-RangeMin	ISO num 范围的最小值
	ISONumRange-Max	ISO num 范围的最大值
	AdjustTarget-Max	各 LV 的 target max50,50, 50, 50, 50, 50, 55, 60, 60, 60, 60, 60, 60, 65, 70, 70
	AdjustTarget-Min	各 LV 的 target min40,40, 40, 40, 40, 40, 45, 50, 50, 50, 50, 50, 50, 55, 60, 60
ExposureInfo	ExposureTime	当前曝光时间, 单位为微秒 (us)。FSWDR 模式下, 表示当前最短帧 (VS) 曝光时间。
	ShortExpTime	FSWDR 模式下, 表示当前短帧 (S) 曝光时间, 单位为微秒 (us)。线性模式不用关注该值。
	LongExpTime	FSWDR 模式下, 表示当前长帧 (L) 曝光时间, 单位为微秒 (us)。线性模式不用关注该值。
	AGain	当前 sensor 模拟增益, 10bit 小数精度。
	DGain	当前 sensor 数字增益, 10bit 小数精度。
	ISPDGain	当前 ISP 数字增益, 10bit 小数精度
	AveLum	平均亮度信息。
	ISO	当前 sensor 模拟增益 *sensor 数字增益 *ISP 数字增益 *100, 其中增益的精度都为 10bit。
	WDRShort-AveLuma	WDR 短帧画面平均亮度信息。
	WDRExpRatio	WDR 长, 短帧的曝光比
	LEFrameAvgLuma	WDR 长帧画面的平均亮度
	SE-FrameAvgLuma	WDR 短帧画面的平均亮度
	LightValue	估计的环境亮度值
AE Route	TotalNum	曝光分配路线节点数目, 目前最大为 16
	AERouteNodeX.IntTime	曝光分配路线节点属性的节点曝光时间, 单位为微秒 (us)
	AERouteNodeX.SysGain	曝光分配路线节点属性的节点增益, 包括 sensor 模拟增益, sensor 数字增益和 ISP 数字增益, 10bit 精度。
	AERouteNodeX.IrisFNO	曝光分配路线节点属性的节点光圈 F 值大小, 仅支持 P-Iris, 不支持 DC-Iris
	AERouteNodeX.IrisFNOLin	曝光分配路线节点属性的节点光圈 F 值等效增益大小, 仅支持 P-Iris, 不支持 DC-Iris
WBAttr	ByPass	白平衡模块 Bypass 使能, 默认值 CVI_FALSE
	AWBRunInterval	白平衡模块工作频率
	OpType	自动白平衡和手动白平衡切换

下页继续

表 43.1 – 续上页

参数	描述
AlgType	ALG_AWB ALG_AWB_SPEC
DebugMode	debug mode
Rgain	手动白平衡红色通道增益
Grgain	手动白平衡 Gr 通道增益
Gbgain	手动白平衡 Gb 通道增益
Bgain	手动白平衡 Gb 通道增益
Enable	自动白平衡使能
RefColorTemp	静态白平衡系数标定的环境色温, 单位 Kelvin。推荐在 Macbeth D50 标准光源环境或室外晴天环境取得 24 色卡 Raw 数据进行标定
RGStrength	自动白平衡 R 通道校准强度
BGStrength	自动白平衡 B 通道校准强度
Speed	自动白平衡算法收敛速度
ZoneSel	参数为 0 或 255 时, 采用近似灰世界的白平衡算法, 其他值则为进行分类筛选, 提升精度
HighColorTemp	自动白平衡算法的色温上限
LowColorTemp	自动白平衡算法的色温下限
ShiftLimitEn	AWB 超过类白点范围的增益映像回白点范围的开关
GainNormEn	对 RGB 通道增益进行限制, 可以改善低色温、低照度场景的信噪比
NaturalCastEn	低色温下 AWB 风格喜好开关
AWBZoneWtEn	画面分区权重开关, 预设为关闭
stCTLimit.bEnable	定义自平衡的增益范围限制属性
stCTLimit.enOpType	Auto Mode
stCTLimit.HighRgain	手动模式下高色温下的最大 R 增益
stCTLimit.HighBgain	手动模式下高色温下的最小 B 增益
stCTLimit.LowRgain	手动模式下低色温下的最小 R 增益
stCTLimit.LowBgain	手动模式下低色温下的最大 B 增益
stLumaHist.bEnable	AW B 亮度与权重参数开关
stLumaHist.enOpType	Auto Mode
stCbCr-Track.bEnable	AWB 统计范围与 ISO 的连动参数
Tolerance	自动白平衡调整的偏差范围, 检测误差在门限范围内时,AWB 不作动作
ZoneRadius	自动白平衡统计中对像素分类时用的距离范围。该值越小,AWB 精度越高, 但会降低 AWB 算法稳定性
CurveLLimit	自动白平衡色温曲线的左边界限
CurveRLimit	自动白平衡色温曲线的右边界限
ExtraLightEn	自动白平衡计算时是否考虑色温曲线外的独立光源点, 最多四个独立点
WhiteBgain	特殊光源点的 B 通道增益
ExpQuant	根据外在亮度做判断。ExpQuant 为开启的亮度限制值 例如 ExpQuant = 6, 表示 LV6 以下开启此 WB 光源点 (一般夜景为 LV6 以下) ExpQuant =106 表示 LV6 以上开启 ExpQuant =112 表示 LV12 以上开启 (LV12 一般为户外)

下页继续

表 43.1 – 续上页

参数		描述
	LightStatus	特殊光源点的种类, 0: 不作动作 1: 加入光源点 2: 删除光源点附近的计算
	Radius	特殊光源点的区域大小,
	OutdoorStatus	室内或室外模式 (手动模式下)
	OutThresh	判定室内室外的阈值, 亮度小于时, 则判定为室内, 户外 LV 大多超过 15
	LowStart	将高色温的权重拉低, 高色温区的起始点, 建议为 6500K
	LowStop	将低色温的权重拉低, 低色温区的终止点, 建议为 4500K
	HighStart	将高色温的权重拉低, 高色温区的起始点, 建议为 6500K
	HighStop	将高色温的权重拉低, 高色温区的终止点, 建议为 8000K
	GreenEnhanceEn	在绿色植物场景下, 对绿色通道增加的开关
	OutShiftLimit	当判定为户外场景时,AWB 算法的白点范围限制
	MultiLightSourceEn	AWB 检测当前场景是否为混合光源, 来调整饱和度或 CCM
	MultiLSType	调整饱和度或是 CCM
	MultiLSScaler	当混合光源下调整饱和度或 CCM 的强度
	FineTunEn	AWB 特殊色检测开关, 例如肤色
	FineTunStrength	肤色、蓝色等特殊色检测的强度
	ShiftLimit	当判定为户外场景时,AWB 算法的白点范围限制
	CurvePara	CurvePara 【0-2】普朗克曲线系数, 由 AWB 标定工具给出。普朗克曲线描绘白色块在不同色温的标准光源下的颜色表现。CurvePara 【3-5】色温曲线系数, 由 AWB 标定工具给出。色温曲线描绘白色块的颜色表现与色温的对应关系。
	StaticWB	静态白平衡系数, 由 AWB 标定工具给出。
	AttrZoneWt	32x32 画面权重
	stLumaHist.HistTh	亮度分类的阈值 (手动模式下有效)
	stLumaHist.HistWt	亮度分类的权重 (手动模式下有效)
	stCbCrTrack.CrMax	不同 ISO 下 R/G 的最大值
	stCbCrTrack.CrMin	不同 ISO 下 R/G 的最小值
	stCbCrTrack.CbMax	不同 ISO 下 B/G 的最大值
	stCbCrTrack.CbMin	不同 ISO 下 B/G 的最小值
	MultiCTBin	色温分段参数
	MultiCTWt	色温分段权重
WBInfo	Rgain	当前 R 通道增益值
	Ggain	当前 G 通道增益值
	Bgain	当前 B 通道增益值
	Saturation	当前饱和度值
	ColorTemp	当前色温值
	InOutStatus	室内外检测结果

下页继续



表 43.1 – 续上页

参数		描述
	Bv	当前环境 bv 值
	CCM	当前颜色校正矩阵值, 8bit 小数精度。bit15 是符号位, 0 表示正数, 1 表示负数, 例如 0x8010 表示-16。
BNR	Enable	BNR 模块使能
	isManualMode	手动模式或自动模式
	WindowType	去噪滤波局域程度。其值越小, 作用越局域。
	DetailSmooth-Mode	去噪细节平滑功能使能。
	NoiseSup-pressStr	噪声抑制强度。值越大, 亮噪去除强度越大。
	FilterType	去噪滤波器强度。值越大, 亮噪去除强度越大。
	NrLscRatio	参考 LSC 增益调节去噪强度。值越大, 参考 LSC 增益的比例越多。
	VarThr	侦测边缘的阈值。值越大, 判断为边缘的数量越少。
	CoringWgtLF	调节在低频区域的随机噪声强度。值越大, 在低频区域保留的噪声越多。
	CoringWgtHF	调节在高频区域的随机噪声强度。值越大, 在高频区域的保留的噪声越多。
	NonDirFiltStr	调节在低频区的去噪强度。值越大, 在低频区域去除的噪声越多。
	VhDirFiltStr	调节在水平和垂直区的去噪强度。值越大, 在水平和垂直边缘去除的噪声越多。
	AaDirFiltStr	调节在对角线边缘的去噪强度。值越大, 在对角线边缘去除的噪声越多。
Crosstalk	Enable	GE 模块使能
	isManualMode	手动模式或自动模式
	Strength	G 通道平衡全局强度
	FlatThre	平坦区侦测节点 1-4 阈值
	GrGbD-iffThreSec	G 通道平衡节点 1-4 阈值
Demosaic	Enable	Demosaic 模块使能
	LumaTuned-CoringEn	(OpType, CoringEn) (x,0): by noise profile (1,1): by ISO table (0,1): by manual
	isManualMode	手动模式或自动模式
	CoarseEdgeThr	边缘粗调侦测阈值。值越小, 侦测为边缘的数量越多。建议搭配参数 CoarseStr 调试。
	CoarseStr	边缘粗调强度值。值越小, 越偏方向性的处理。反之, 越偏无方向性的处理。
	FineEdgeThr	边缘细调侦测阈值。值越小, 侦测为边缘的数量越多。建议搭配参数 FineStr 调试。
	FineStr	边缘细调强度值。值越小, 越偏方向性的处理。反之, 越偏无方向性的处理。
	De-tailSmoothEn-able	细节平滑功能使能

下页继续

表 43.1 – 续上页

参数		描述
	DetailSmooth-Str	细节平滑强度。值越大，平滑强度越强，对伪细节的抑制强度越大。
	DetailWgtThr	细节保留范围阈值。值越小，细节保留作用的范围越大。
	DetailWgtSlope	细节保留强度。值越大，细节保留越多。
	DetailWgtMin	边缘细节平滑保留允许之最小增益。
	DetailWgtMax	边缘细节平滑保留允许之最大增益。
	LumaWgtThr	亮度细节平滑作用范围阈值。值越小，细节平滑作用的范围越大。
	LumaWgtSlope	亮度细节平滑边缘保留强度。值越大，细节平滑强度越强。
	LumaWgtMin	亮度细节平滑强度允许之最小增益。
	LumaWgtMax	亮度细节平滑强度允许之最大增益。
	EdgeEnhanceEnable	边缘强化功能使能
	OverShtGain	调节 overshoot 程度。值越大，边缘强化越大。OverShtGain 为一倍时，其值为 256。
	UnderShtGain	调节 undershoot 程度。值越大，边缘强化越大。UnderShtGain 为一倍时，其值为 256。
	NoiseSuppressStr	噪声抑制强度。值越大，越能抑制噪声被强化。
	GainTable	无方向中频纹理增强强度。值越大，无方向中频纹理的锐度越高。该参数是一个 33 的数组，表现为一个连续 33 段的强度曲线。数值为 128 时，增益为一倍。
	LumaTuned-Coring	噪声抑制强度。值越大，越能抑制噪声被强化。
AE config	WinWeight	AE 测光权重

### 43.4.2 LEVEL2 级别调试信息分析

在 level1 的基础上，补充了 meshLscGainLut，exposureInfo 的额外资讯，以及 GammaTable

#### 【参数说明】

参数		描述
Module/GammaEx	GammaTable	Gamma 曲线节点数值
ShadingEx	MeshLscGain-Lut	LSC 网格线形式红色通道补偿增益表, 包含 R G B 三个通道
ExposureInfoEx	Exposure	当前曝光量, 等于曝光时间与曝光增益的乘积, 其中曝光时间的单位为曝光行数, 曝光增益为 6bit 小数精度。
	ExposureIsMax	0: ISP 未达到最大曝光水平; 1: ISP 达到最大曝光水平。
	HistError	统计信息, AE 的目标亮度值与实际值的差该值为正表示当前期望的亮度信息大于实际的亮度信息, 该值为负表示期望的亮度信息小于实际的亮度信息。
	PirisFno	当前 P-Iris 光圈 F 值对应的等效增益。
	Fps	实际图像帧率 * 100。
	RefExpRatio	参考曝光比, 用于估计当前场景的动态范围, 会受到 ISP_WDR_EXPOSURE_ATTR_S 中 Tolerance 和 Speed 等值的影响。
	AE_RouteTotalNum	曝光分配路线节点数目, 目前最大为 16
	AERouteNodeX.IntTime	曝光分配路线节点属性的节点曝光时间, 单位为微秒(us)
	AERouteNodeX.SysGain	曝光分配路线节点属性的节点增益, 包括 sensor 模拟增益, sensor 数字增益和 ISP 数字增益, 10bit 精度。
	AERouteNodeX.IrisFNO	曝光分配路线节点属性的节点光圈 F 值大小, 仅支持 P-Iris, 不支持 DC-Iris
	AE_RouteNodeX.IrisFNOLin	曝光分配路线节点属性的节点光圈 F 值等效增益大小, 仅支持 P-Iris, 不支持 DC-Iris
	AE_Hist256Value	全局 256 段直方图统计信息

### 43.4.3 LEVEL3 级别调试信息分析

在 level2 的基础上, 补充了 3a 的统计值

#### 【参数说明】

参数		描述
AE statistics	LEGlobalAvgR	长曝光帧全局统计的 R 分量统计平均值
	LEGlobalAvgGr	长曝光帧全局统计的 GR 分量统计平均值
	LEGlobal-AvgGb	长曝光帧全局统计的 GB 分量统计平均值
	LEGlobalAvgB	长曝光帧全局统计的 B 分量统计平均值
	SEGlobalAvgR	短曝光帧全局统计的 R 分量统计平均值
	SEGlobalAvgGr	短曝光帧全局统计的 GR 分量统计平均值
	SEGlobalAvgGb	短曝光帧全局统计的 GB 分量统计平均值
	SEGlobalAvgB	短曝光帧全局统计的 B 分量统计平均值
	LEHistogram-Mem	长曝光帧全局统计的像素 histogram 统计值

下页继续



表 43.2 – 续上页

参数		描述
	SEHistogram-Mem	短曝光帧全局统计的像素 histogram 统计值
	LEZoneRAvg	长曝光帧各分区间 R 通道的分量统计平均值
	LEZoneGrAvg	长曝光帧各分区间 Gr 通道的分量统计平均值
	LEZoneGbAvg	长曝光帧各分区间 Gb 通道的分量统计平均值
	LEZoneBAvg	长曝光帧各分区间 B 通道的分量统计平均值
	SEZoneRAvg	短曝光帧各分区间 R 通道的分量统计平均值
	SEZoneGrAvg	短曝光帧各分区间 Gr 通道的分量统计平均值
	SEZoneGbAvg	短曝光帧各分区间 Gb 通道的分量统计平均值
	SEZoneBAvg	短曝光帧各分区间 B 通道的分量统计平均值
AWB statistics	LEGlobalR	长曝光帧全局统计的 R 分量统计平均值
	LEGlobalG	长曝光帧全局统计的 G 分量统计平均值
	LEGlobalB	长曝光帧全局统计的 B 分量统计平均值
	LECountAll	长曝光帧全局统计的像素统计平均值
	SEGlobalR	短曝光帧全局统计的 R 分量统计平均值
	SEGlobalG	短曝光帧全局统计的 G 分量统计平均值
	SEGlobalB	短曝光帧全局统计的 B 分量统计平均值
	SECountAll	短曝光帧全局统计的像素统计平均值
	LEZoneAvgR	长曝光帧各分区间的 R 分量统计平均值
	LEZoneAvgG	长曝光帧各分区间的 G 分量统计平均值
	LEZoneAvgB	长曝光帧各分区间的 B 分量统计平均值
	LEZoneCountAll	长曝光帧各分区间的像素统计值
	SEZoneAvgR	短曝光帧各分区间的 R 分量统计平均值
	SEZoneAvgG	短曝光帧各分区间的 G 分量统计平均值
	SEZoneAvgB	短曝光帧各分区间的 B 分量统计平均值
	SEZoneCountAll	曝光帧各分区间的像素统计值

# 44 3A 开发指南概述

---

# 45 3A 开发用户指南

## 45.1 AF 统计信息使用说明

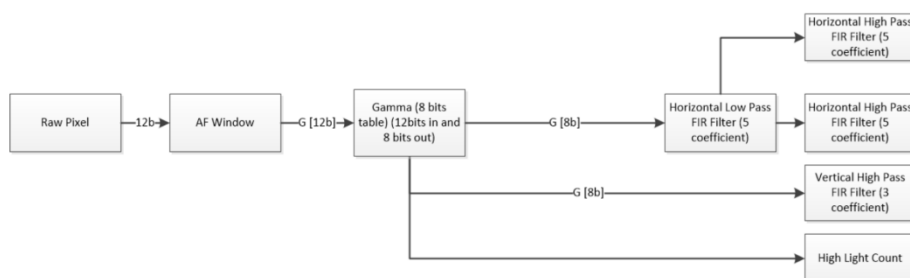
### 45.1.1 概述

自动对焦是通过分析图像的特征得到目前图像清晰值 FV(Focus Value), 由于图像越清晰 FV 越大, 因此经过比对每个位置的清晰值可以找出 FV 值曲线的最高点, 此位置即为对焦点, 找出对焦点后使用者即可控制对焦马达达到最佳位置完成自动对焦。

目前 AF 共提供四个滤波器与亮度讯息, 分别为水平方向 H1, H2, 垂直方向 V1, 其中水平方向会先经过一个低通滤波器后再分别经过两个高通滤波器以得到 H1,H2 的值, 垂直方向则只经过高通滤波器,

目前 AF 只支持 bayer 域的统计信息

#### Sequence



### 45.1.2 输入图像的裁剪

AF 支持对输入图像的裁剪, 用户可通过 stCrop 中的 X, Y, W, H 来决定目前 AF 的统计区域, 详情可以参考 ISP\_STATISTICS\_CFG\_S 以获得相关结构信息

### 45.1.3 Bayer 域的配置

数据在进 AF 模块前有两样前处理可以使用

1. 使用者可选择是否需要经过 gamma, 假如需要的话, 需使能 gamma 并填入适当的 gamma 曲线
2. 使用者可选择是否开启预滤波处理以消除椒盐噪声对于统计值的影响

### 45.1.4 抑制光源对于 FV 值的影响

在图像中具有点光源的情况, 在对焦模糊时会因为光晕扩散影响到 FV 值, 而发生图像模糊 FV 值却增大的状况, 为了抑制此现象, 增加 u16HlCnt 来统计窗口中高亮度点的数值, 使用者可调整 u16HighLumaTh 来决定高亮点的阈值, 当模糊时窗口中的亮点数量增加, 清晰时窗口中的亮点个数最少, 用户可使用此信息来判断最适合的对焦点

### 45.1.5 统计信息配置注意事项

类型	描述
分块大小	最大 7 * 15
统计模块工作域	RAW
Bayer 统计参数是否减去黑电平	已减去

### 45.1.6 FV 值的获取

当图像的最后一个 pixel 通过后即可取得统计信息, 使用者可通过 CVI\_ISP\_GetVDTimeOut 以同步获得统计值, 可参考 33.1.8 的流程

### 45.1.7 FV 值的计算

一个 block 可取得的统计值有三种, 分别为 Horizontal 第一组 filter 得到的 H0, Horizontal 第二组 filter 得到的 H1, Vertical filter 得到的 V0, 我们将每个 block 的 FV 值命名为 FV<sub>n</sub>, 为每个统计值设置自己的权重, 分别为 W0 / W1 / W2, 则 FV<sub>n</sub> 的值为

$$FV_n = \frac{W_0 * H0_n + W_1 * H1_n + W_2 * V0_n}{W_0 + W_1 + W_2}$$

而最终的 FV 值还需要为每个 block 的 FV 加上权重, 假设第 n 个 block 的 weight 为 W<sub>n</sub>, 最终 FV 的值则为:

$$FV = \frac{\sum_{n=0}^{blocks} FV_n * W_n}{\sum_{n=0}^{blocks} W_n}$$

### 45.1.8 FV 计算参考代码

```

ISP_AF_STATISTICS_S afStat;
CVI_U32 row, col;
CVI_S32 s32Ret = CVI_SUCCESS;
CVI_U64 stsValue = 0;
VI_PIPE ViPipe = 0;
ISP_VD_TYPE_E enIspVDType = ISP_VD_FE_START;
CVI_CHAR input[10];
ISP_STATISTICS_CFG_S stsCfg;
ISP_PUB_ATTR_S stPubAttr;
struct timeval t1, t2;
// AF weighting table
static int AFWeight[15][17] = {{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                                {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}};

// Get current statistic and related size setting.
s32Ret = CVI_ISP_GetStatisticsConfig(ViPipe, &stsCfg);
s32Ret |= CVI_ISP_GetPubAttr(ViPipe, &stPubAttr);
if (s32Ret != CVI_SUCCESS) {
    CVI_TRACE_LOG(CVI_DBG_ERR, "Get Statistic info fail with %s!\n", s32Ret);
    return s32Ret;
}
// Config AF Enable.
stsCfg.stFocusCfg.stConfig.bEnable = 1;
// Config low pass filter.
stsCfg.stFocusCfg.stConfig.u8HFiltShift = 0;
stsCfg.stFocusCfg.stConfig.s8HVFiltLpCoeff[0] = 0;
stsCfg.stFocusCfg.stConfig.s8HVFiltLpCoeff[1] = 1;
stsCfg.stFocusCfg.stConfig.s8HVFiltLpCoeff[2] = 2;
stsCfg.stFocusCfg.stConfig.s8HVFiltLpCoeff[3] = 3;
stsCfg.stFocusCfg.stConfig.s8HVFiltLpCoeff[4] = 4;
// Config gamma enable.

```

(下页继续)

(续上页)

```

stsCfg.stFocusCfg.stConfig.stRawCfg.PreGammaEn = 0;
// Config pre NR enable.
stsCfg.stFocusCfg.stConfig.stPreFltCfg.PreFltEn = 1;
// Config H & V window.
stsCfg.stFocusCfg.stConfig.u16Hwnd = 17;
stsCfg.stFocusCfg.stConfig.u16Vwnd = 15;
// Config crop related setting. Has some limitation
stsCfg.stFocusCfg.stConfig.stCrop.bEnable = 1;
stsCfg.stFocusCfg.stConfig.stCrop.u16X = 8;
stsCfg.stFocusCfg.stConfig.stCrop.u16Y = 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16W = stPubAttr.stWndRect.u32Width - 8 * 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16H = stPubAttr.stWndRect.u32Height - 2 * 2;
// Config first horizontal high pass filter.
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[1] = -3;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[2] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[3] = -10;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[4] = 0;
// Config 2nd horizontal high pass filter.
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[1] = -3;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[2] = 0;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[3] = -10;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[4] = 0;
// Config vertical high pass filter.
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[0] = 8;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[1] = -15;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[2] = 0;
stsCfg.unKey.bit1FEAfStat = 1;
s32Ret = CVI_ISP_SetStatisticsConfig(ViPipe, &stsCfg);
if (s32Ret != CVI_SUCCESS) {
    CVI_TRACE_LOG(CVI_DBG_ERR, "ISP Set Statistic failed with %#x!\n", s32Ret);
    return s32Ret;
}

printf("select. c -> Fv curve\n");
printf("..... h0 -> print h0 blocks statistic\n");
printf("..... h1 -> print h1 blocks statistic\n");
printf("..... v0 -> print v0 blocks statistic\n");
printf("..... hlc -> print hlcnt blocks statistic\n");
scanf("%s", input);

while(1) {
    // Wait VD start for get focus statistic data.
    s32Ret = CVI_ISP_GetVDTimeOut(ViPipe, enIsPVDType, 5000);
    s32Ret |= CVI_ISP_GetFocusStatistics(ViPipe, &afStat);
    if (s32Ret != CVI_SUCCESS) {
        CVI_TRACE_LOG(CVI_DBG_ERR, "Get Statistic failed with %#x!\n", s32Ret);
        return CVI_FAILURE;
    }
    // print each focus statistic.
    if (strcmp(input, "c", 1) != 0) {
        for (row = 0; row < AF_ZONE_ROW; row++) {
            for (col = 0; col < AF_ZONE_COLUMN; col++) {

```

(下页继续)

(续上页)

```

        if (strncmp(input, "h0", 2) == 0) {
            stsValue = afStat.stFEAFStat.stZoneMetrics[row][col].u64h0;
        } else if (strncmp(input, "h1", 2) == 0) {
            stsValue = afStat.stFEAFStat.stZoneMetrics[row][col].u64h1;
        } else if (strncmp(input, "v0", 2) == 0) {
            stsValue = afStat.stFEAFStat.stZoneMetrics[row][col].u32v0;
        } else {
            stsValue = afStat.stFEAFStat.stZoneMetrics[row][col].u16HlCnt;
        }
        printf("%d ", stsValue);
    }
    printf("\n");
}
continue;
}

CVI_U64 FVn = 0, FV = 0;
CVI_U32 totalWeightSum = 0;
// weight for each statistic
const CVI_U32 weight1 = 1, weight2 = 1, weight3 = 1;
const CVI_U32 blockWeightSum = weight1 + weight2 + weight3;
// calculate AF statistics
for (row = 0; row < AF_ZONE_ROW; row++) {
    for (col = 0; col < AF_ZONE_COLUMN; col++) {
        CVI_U64 h0 = afStat.stFEAFStat.stZoneMetrics[row][col].u64h0;
        CVI_U64 h1 = afStat.stFEAFStat.stZoneMetrics[row][col].u64h1;
        CVI_U32 v0 = afStat.stFEAFStat.stZoneMetrics[row][col].u32v0;

        FVn = (weight1 * h0 + weight2 * h1 + weight3 * v0) / blockWeightSum;
        FV += FVn * AFWeight[row][col];
        totalWeightSum += AFWeight[row][col];
    }
}
FV = FV / totalWeightSum;
}
return CVI_SUCCESS;

```

# 46 开发者指南

## 46.1 概述

当图像的最后一个人像通过后即可取得统计信息，使用者可通过 CVI\_ISP\_GetVDTimeOut 以同步获得统计值，可参考 33.1.8 的流程。

因为 Linux 系统 user space 任务调度不能保证一致的实时性，需要将驱动配置放在 kernel space 完成。

ISP 提供同步回调接口的注册，可以实现与 VD 同步。

在本章有相应的接口使用描述，用户可以将实时性要求较高的任务放在同步回调里面，底层提供 HwIRQ，Workqueue 两种方式实现，可以选择相应的实现方式以确定实时级别。

HwIRQ 是指任务放在中断服务中实现，实时性最高，Workqueue 的实时性取决于 linux 系统调用。

## 46.2 API 参考

- `isp_sync_task_register`：向 ISP 注册同步回调接口。
- `isp_sync_task_unregister`：向 ISP 反注册同步回调接口。

### 46.2.1 `isp_sync_task_register`

#### 【描述】

向 ISP 注册同步回调接口

#### 【语法】

```
int isp_sync_task_register(int vi_pipe, struct isp_sync_task_node *new_node);
```

#### 【参数】



参数名称	描述	输入/输出
vi_pipe	VI_PIPE 号	输入
new_node	新插入的同步回调节点	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败。

**【需求】**

- 头文件: `cvi_vip_isp_ext.h`
- 库文件:

**【注意】**

- 使用前需要确保 ISP 驱动已加载。
- 因为 ISP 同步回调内部实现不保存用户传入的 `new_node` 指向的实体，所以要求使用 `isp_sync_task_node` 定义实体时不能为局部变量。

**【举例】**

无。

**【相关主题】**

- `isp_sync_task_unregister`

## 46.2.2 isp\_sync\_task\_unregister

**【描述】**

向 ISP 反注册同步回调接口

**【语法】**

```
int isp_sync_task_unregister(int vi_pipe, struct isp_sync_task_node *del_node);
```

**【参数】**

参数名称	描述	输入/输出
vi_pipe	VI_PIPE 号	输入
del_node	需要删除的同步回调节点	输入

**【返回值】**

返回值	描述
0	成功。
非 0	失败。

【需求】

- 头文件: `cvi_vip_isp_ext.h`
- 库文件:

【注意】

- 使用前需要确保 ISP 驱动已加载。

【举例】

无。

【相关主题】

- `isp_sync_task_register`

## 46.3 数据类型

- `isp_sync_tsk_method`：定义同步回调方法，决定实时性。
- `isp_sync_task_node`：定义同步回调节点信息。

### 46.3.1 `isp_sync_tsk_method`

【说明】

定义同步回调方法，决定实时性

【定义】

```
enum isp_sync_tsk_method {
    ISP_SYNC_TSK_METHOD_HW_IRQ = 0,
    ISP_SYNC_TSK_METHOD_WORKQUEUE,
    ISP_SYNC_TSK_METHOD_BUTT
};
```

【成员】

成员名称	描述
<code>ISP_SYNC_TSK_METHOD_HW_IRQ</code>	使用硬件中断方式回调。
<code>ISP_SYNC_TSK_METHOD_WORKQUEUE</code>	使用工作队列方式回调。

【注意事项】

无。

【相关数据类型及接口】

无。

## 46.3.2 isp\_sync\_task\_node

### 【说明】

定义同步回调节点信息

### 【定义】

```
struct isp_sync_task_node {  
    enum isp_sync_tsk_method method;  
    __s32 (*isp_sync_tsk_call_back)(__u64 data);  
    __u64 data;  
    const char *sz_id;  
    struct list_head list;  
};
```

### 【成员】

成员名称	描述
method	回调方式。
isp_sync_tsk_call_back	回调函数，用户注册时传入。
data	回调函数参数，用户注册时传入。
sz_id	节点 ID。
list	list 节点，用于管理多个回调节点，无需关注。

### 【举例】

```
isp_sync_task_node sync_node = {  
    .method = ISP_SYNC_TSK_METHOD_HW_IRQ,  
    .isp_sync_tsk_call_back = sync_af_calc,  
    .data = 0,  
    .sz_id = "hw_0"  
};
```

### 【注意事项】

无。

### 【相关数据类型及接口】

- `isp_sync_tsk_method`

# 47 附录

---

# 48 缩略语

---