SDK6 AN A12 IQ Parameter

Draft Version 0.2

September 16, 2015

Confidentiality Notice:

The contents of this document are proprietary and confidential information of Ambarella Inc.

The material in this document is for information only. Ambarella assumes no responsibility for errors or omissions and reserves the right to change, without notice, product specifications, operating characteristics, packaging, ordering, etc. Ambarella assumes no liability for damage resulting from the use of information contained in this document. All brands, product names and company names are trademarks of their respective owners.

US

3101 Jay Street
Ste. 110
Santa Clara, CA 95054, USA
Phone: +1.408.734.8888
Fax: +1.408.734.0788

Hong Kong

Unit A&B, 18/F, Spectrum Tower
53 Hung To Road
Kwun Tong, Kowloon
Phone: +85.2.2806.8711
Fax: +85.2.2806.8722

Korea

6 Floor, Hanwon-Bldg.
Sunae-Dong, 6-1, Bundang-Gu
SeongNam-City, Kyunggi-Do
Republic of Korea 463-825
Phone: +031.717.2780
Fax: +031.717.2782

China - Shanghai

9th Floor, Park Center
1088 Fangdian Road, Pudong New District
Shanghai 201204, China
Phone: +86.21.6088.0608
Fax: +86.21.6088.0366

Taiwan

Suite C1, No. 1, Li-Hsin Road 1
Science-Based Industrial Park
Hsinchu 30078, Taiwan
Phone: +886.3.666.8828
Fax: +886.3.666.1282

Japan - Yokohama

Shin-Yokohama Business Center Bldg. 5th Floor
3-2-6 Shin-Yokohama, Kohoku-ku,
Yokohama, Kanagawa, 222-0033, Japan
Phone: +81.45.548.6150
Fax: +81.45.548.6151

China - Shenzhen

Unit E, 5th Floor
No. 2 Finance Base
8 Ke Fa Road
Shenzhen, 518057, China
Phone: +86.755.3301.0366
Fax: +86.755.3301.0966

I Contents

II Preface

This document provides technical details using a set of consistent typographical conventions to help the user differentiate key concepts at a glance.

II.1 Typographical Conventions

Conventions include:

Example	Description
AmbaGuiGen, **DirectUSB** **Save**, **File > Save** **Power**, **Reset**, **Home**	Software names GUI commands and command sequences Computer / Hardware buttons
Flash_IO_control **da**, **status**, **enable**	Register names and register fields. For example, **Flash_IO_control** is the register for global control of Flash I/O, and bit 17 (**da)** is used for DMA acknowledgement.
GPIO81, **CLK_AU**	Hardware external pins
VIL, VIH, VOL, VOH	Hardware pin parameters
INT_O, RXDATA_I	Hardware pin signals
amb_performance_t ***amb_operating_mode_t*** **amb_set_operating_mode()**	API details (e.g., functions, structures, and type definitions)

`/usr/local/bin`
`success = amb_set_operating_ mode (amb_hal_base_address, & operating_mode)`
	User entries into software dialogues and GUI windows
	File names and paths
	Command line scripting and Code

Table II-1. Typographical Conventions for Technical Documents.

Additional Ambarella typographical conventions include:

- Acronyms are given in UPPER CASE using the default font (e.g., AHB, ARM11 and DDRIO).
- Names of Ambarella documents and publicly available standards, specifications, and databooks appear in *italic* type.

II.2 Abbreviations

Acronym	Definition
AAA Function	AE, AWB and AF function
AE	Automatic Exposure
AF	Auto Focus
AGC	Automatic Gain Control
AWB	Automatic White Balance
BLC	Black Level Correction
CC	Color Correction
CDNR	Color Dependent Noise Reduction
CFA	Color Filter Array
EIS	Electronic Image Stabilization
EV	Exposure Value
EXIF	Exchange Image file format
FIR	Finite Impulse Response
FPS	Frames per second
IDSP	Integrated Digital Signal Processing
IQ	Image Quality
IS	Image Stabilization
ISO	International Organization for Standarization
LUT	Look Up Table
LV	Luma value
MCTF	Name of Filter block in Ambarella IDSP
Nf-index	Noise Filter Index
OR	OR function (where “0 OR 0 is 0” and “0 OR 1 is 1”...)
ROI	Region of Interest
SDK	System Design Kit
WB	White Balance

1 Overview

1.1 Overview: Introduction

This document is a guide to A12 SDK image quality (IQ) tuning and covers all aspects including automatic exposure (AE), automatic white balance (AWB), color tuning, video tuning, and still-picture tuning. It includes details on how to tune the digital effect and use the scene mode control.

The image quality tuning details are included in

- Chapter 2 “Image Quality (IQ) Tuning”

2 Image Quality (IQ) Tuning

2.1 IQ Tuning: Introduction

This document covers issues related to tuning image quality (IQ) and is split into the following sections:

- (Section 2.1) IQ Tuning: Introduction
- (Section 2.2) IQ Tuning: Data Structures
- (Section 2.3) IQ Tuning: Header File Details
- (Section 2.4) IQ Tuning: Default Binary Color Table Files

There are five kinds of header files in the A12 system design kit (SDK) that are used to store image quality (IQ) settings:

1. (Section 2.3.1) Header File Details: AmbaIQParamXXX_A12_DefaultParams.c
2. (Section 2.3.2) Header File Details: AmbaIQParamXXX_A12_ADJ_VideoXXX.c
   (Section 2.3.3) Header File Details: AmbaIQParamXXX_A12_ADJ_StillLIso.c
   (Section 2.3.4) Header File Details: AmbaIQParamXXX_A12_ADJ_StillHIso.c
3. (Section 2.3.5) Header File Details: AmbaIQParamXXX_A12_ImageParam.c
4. (Section 2.3.6) Header File Details: AmbaIQParamXXX_A12_ScXXXParam.c
5. (Section 2.3.7) Header File Details: AmbaIQParamXXX_A12_DeXXXParam.c

AmbaIQParamXXX_A12_DefaultParams.c can be modified to set the AE/AWB related parameters.

AmbaIQParamXXX_A12_Adj_XXX.c, can be modified to set the auto-adjust parameters, including the AWB ratio, AE target, and some noise/sharpness filters.

AmbaIQParamXXX_A12_ImageParam.c can be modified to set the RGB2YUV matrix and some initial parameters.

AmbaIQParamXXX_A12_ScXXXParam.c can be modified to set the scene parameters.

AmbaIQParamXXX_A12_DeXXXParam.c can be modified to set the digital effect parameters.

In addition to the five parameter files, several default binaries need to be loaded for IQ tuning. These binary files are listed in

- (Section 2.4) IQ Tuning: Default Binary Color Table Files.

2.2 IQ Tuning: Data Structures

This section provides information on the data structures used for IQ tuning in the A12 SDK. For more details regarding these data structures, please refer to the header files AmbaImg_AaaDef.h and AmbaImg_Adjustment_A12.h.

- (Section 2.2.1) Data Structures: AAA_PARAM_s

- (Section 2.2.2) Data Structures: ADJ_VIDEO_PARAM_s
- (Section 2.2.3) Data Structures: ADJ_PHOTO_PARAM_s
- (Section 2.2.4) Data Structures: ADJ_STILL_FAST_LISO_PARAM_S
- (Section 2.2.5) Data Structures: ADJ_STILL_HISO_PARAM_S
- (Section 2.2.6) Data Structures: IMG_PARAM_s
- (Section 2.2.7) Data Structures: SCENE_DATA_s
- (Section 2.2.8) Data Structures: DE_PARAM_s
- (Section 2.2.9) Data Structures: CALIBRATION_PARAM_s

2.2.1 Data Structures: AAA_PARAM_s

This structure is defined in Section 2.3.1.1.1.

```
typedef struct _AAA_PARAM_s_ {
  UINT32      StructVersionNum;
  UINT32      ParamVersionNum;
  AMBA_3A_OP_INFO_s AaaFunc;
  AE_CONTROL_s      AeControlMode;
  AE_EV_LUT_s AeEvLut;
  AE_ALGO_INFO_s    AeAlgoInfo;
  AWB_CONTROL_s     AwbControlMode;
  AWB_ALGO_INFO_s   AwbAlgoInfo;
  FLASH_AE_AWB_INFO_s     FlashAeAwbInfo;
  AE_SENSOR_COMPEN_s      SensorCompenInfo;
  AF_CONTROL_s      AfControlMode;
  EIS_CONTROL_s     EisControlMode;
  SCENEMODE_s SceneMode;
} AAA_PARAM_s;
```

2.2.2 Data Structures: ADJ_VIDEO_PARAM_s

This structure is defined in Section 2.3.2.1.1.

```
typedef struct _ADJ_VIDEO_PARAM_s_ {
  UINT32                  VersionNumber;
  UINT32                  ParamVersionNum;
  ADJ_AWB_AE_s            AwbAe;
  VIDEO_FILTER_PARAM_s    FilterParam;
} ADJ_VIDEO_PARAM_s;
```

2.2.3 Data Structures: ADJ_PHOTO_PARAM_s

This structure is defined in Section 2.3.2.1.1.

```
typedef struct _ADJ_PHOTO_PARAM_s_ {
  UINT32                  VersionNumber;
  UINT32                  ParamVersionNum;
  ADJ_AWB_AE_s            NormalAwbAe;
  ADJ_AWB_AE_s            FlashAwbAe;
  VIDEO_FILTER_PARAM_s    FilterParam;
} ADJ_PHOTO_PARAM_s
```

2.2.4 Data Structures: ADJ_STILL_FAST_LISO_PARAM_S

This structure is defined in Section 2.3.3.1.1.

```
typedef struct _ADJ_STILL_FAST_LISO_PARAM_s {

    UINT32      VersionNum;
    UINT32      ParamVersionNum;
    UINT8 NfMaxTableCount;
    ADJ_FILTER_INFO_s NormalEvImg;
    ADJ_FILTER_INFO_s  FlashEvImg;
    ADJ_DEF_s   Def;
    ADJ_BASIC_s Basic;

    ADJ_LUT_AGC_WB_s  ChromaFilter;
    DEF_SHARP_INFO_s  SharpInfo;

} ADJ_STILL_FAST_LISO_PARAM_S;
```

2.2.5 Data Structures: ADJ_STILL_HISO_PARAM_S

This structure is defined in Section 2.3.4.1.1.

```
typedef struct _ADJ_STILL_HISO_PARAM_s_ {

   UINT32      VersionNum;
   UINT32      ParamVersionNum;
   UINT8 NfMaxTableCount;
   ADJ_FILTER_INFO_s NormalEvImg;
   ADJ_FILTER_INFO_s FlashEvImg;
   ADJ_DEF_s   Def;
   ADJ_BASIC_s Basic;

   ADJ_LUT_AGC_WB_s  ChromaFilter;
   DEF_SHARP_INFO_s  SharpInfo;
   ADJ_HISO_FILTER_INFO_s  HIsoNormalEvImg;
   ADJ_HISO_FILTER_INFO_s  HIsoFlashEvImg;
   ADJ_BASIC_s HIsoBasic;

   UINT8 HIsoCdnrEnable;
   ADJ_LUT_s   HIsoCdnrLut[ADJ_HISO_NF_TABLE_COUNT];

   DEF_ASF_INFO_s    HIsoAsf;
   DEF_ASF_INFO_s    HIsoHighAsf;
   DEF_ASF_INFO_s    HIsoMed1Asf;
   DEF_ASF_INFO_s    HIsoMed2Asf;
   DEF_ASF_INFO_s    HIsoLowAsf;
```

```
        DEF_SHARP_s HIsoHighSharp;
        DEF_SHARP_s HIsoMedSharp;
        DEF_SHARP_s HIsoLiSharp;

        DEF_ASF_INFO_s    HIsoChromaAsf;

        ADJ_LUT_AGC_WB_s  HIsoChromaFilterPre;
        ADJ_LUT_AGC_WB_s  HIsoChromaFilterHigh;
        ADJ_LUT_AGC_WB_s  HIsoChromaFilterMed;
        ADJ_LUT_AGC_WB_s  HIsoChromaFilterLow;
        ADJ_LUT_AGC_WB_s  HIsoChromaFilterVLow;
        ADJ_LUT_AGC_WB_s  HIsoChromaFilterLowAndVLow;

        CHROMA_FILTER_COMBINE_s HIsoChromaFilterMedCombine;
        CHROMA_FILTER_COMBINE_s HIsoChromaFilterLowCombine;
        CHROMA_FILTER_COMBINE_s HIsoChromaFilterVLowCombine;

        LUMA_COMBINE_s    HIsoLumaFilterCombine;
        LUMA_COMBINE_s    HIsoLowAsfCombine;

        CHROMA_FILTER_COMBINE_s HIsoLiCombine;

        UINT8 HIsoLiLumaMidHightFreqRcvrEnable;
        DEF_FIR_s   HIsoLiLumaMidHightFreqRcvr;

        UINT8 HIsoLi2ndBlendEnable;
        ADJ_LUT_s   HIsoLi2ndBlend[ADJ_HISO_NF_TABLE_COUNT];

        DEF_ASF_INFO_s    Li2ndAsf;
        DEF_SHARP_s Li2ndSharp;


     } ADJ_STILL_HISO_PARAM_s;
```

2.2.6 Data Structures: IMG_PARAM_s

This structure is defined in Section 2.3.5.1.

```
typedef struct _IMG_PARAM_s_ {
  UINT32                                    VersionNum;
  AMBA_DSP_IMG_BLACK_CORRECTION_s           BlackCorrVideo;
  AMBA_DSP_IMG_BLACK_CORRECTION_s           BlackCorrStill;
  AMBA_DSP_IMG_DBP_CORRECTION_s             BadCorrVideo;
  AMBA_DSP_IMG_DBP_CORRECTION_s             BadCorrStill;
  AMBA_DSP_IMG_CFA_LEAKAGE_FILTER_s         CfaLeakageFilterVideo;
  AMBA_DSP_IMG_CFA_LEAKAGE_FILTER_s         CfaLeakageFilterStill;
  AMBA_DSP_IMG_CFA_NOISE_FILTER_s           CfaNoiseFilterVideo;
  AMBA_DSP_IMG_CFA_NOISE_FILTER_s           CfaNoiseFilterStill;
  AMBA_DSP_IMG_GBGR_MISMATCH_s              GrGbMismatchVideo;
  AMBA_DSP_IMG_GBGR_MISMATCH_s              GrGbMismatchStill;
  AMBA_DSP_IMG_DEMOSAIC_s                   DemosaicVideo;
  AMBA_DSP_IMG_DEMOSAIC_s                   DemosaicStill;

  UINT8                                     AntiAliasingEnableVideo;
  UINT8                                     AntiAliasingEnableStill;
  AMBA_DSP_IMG_AAA_STAT_INFO_s              AaaStatisticsInfo;
  AMBA_DSP_IMG_WB_GAIN_s                    WbGainVideo;
  AMBA_DSP_IMG_WB_GAIN_s                    WbGainStill;
  AMBA_DSP_IMG_LOCAL_EXPOSURE_s             LocalExposureVideo;
  AMBA_DSP_IMG_LOCAL_EXPOSURE_s             LocalExposureStill;

  AMBA_DSP_IMG_COLOR_CORRECTION_s           ColorCorrVideo;
  AMBA_DSP_IMG_COLOR_CORRECTION_s           ColorCorrStill;

  AMBA_DSP_IMG_RGB_TO_YUV_s                 RgbYuvMatrixVideoTv;
  AMBA_DSP_IMG_RGB_TO_YUV_s                 RgbYuvMatrixVideoPc;
  AMBA_DSP_IMG_RGB_TO_YUV_s                 RgbYuvMatrixStill;

  AMBA_DSP_IMG_CHROMA_SCALE_s               ChromaScaleVideo;
  AMBA_DSP_IMG_CHROMA_SCALE_s               ChromaScaleStill;
  AMBA_DSP_IMG_CHROMA_MEDIAN_FILTER_s       ChromaMedianFilterVideo;
  AMBA_DSP_IMG_CHROMA_MEDIAN_FILTER_s       ChromaMedianFilterStill;

  AMBA_DSP_IMG_CHROMA_FILTER_s              ChromaFilterVideo;
  AMBA_DSP_IMG_CHROMA_FILTER_s              ChromaFilterStill;

  UINT16                                    VideoGammaCurve[TONE_CURVE_SIZE];
  UINT16                                    StillGammaCurve[TONE_CURVE_SIZE];
  AMBA_DSP_IMG_DGAIN_SATURATION_s           DGainSaturation;

  AMBA_DSP_IMG_CDNR_INFO_s                  CdnrVideo;
  AMBA_DSP_IMG_CDNR_INFO_s                  CdnrStill;

  AMBA_DSP_IMG_VIDEO_MCTF_INFO_s            MctfInfoVideo;

  AMBA_DSP_IMG_CFA_LEAKAGE_FILTER_s         HIsoCfaLeakageFilter;
  AMBA_DSP_IMG_DEFER_COLOR_CORRECTION_s     HIsoDeferColorCorr;

} IMG_PARAM_s;
```

2.2.7 Data Structures: SCENE_DATA_s

This structure is defined in Section 2.3.6.1.

```
typedef struct _SCENE_DATA_s_ {
  UINT16      SceneMode;
  SCENE_DEF_s Def;
  SCENE_AE_s  AeControl;
  SCENE_AWB_s AwbControl;
  SCENE_ADJ_s AdjControl;
} SCENE_DATA_s;
```

2.2.8 Data Structures: DE_PARAM_s

This structure is defined in Section 2.3.7.1.

```
typedef struct _DE_PARAM_s_ {
  UINT32            VersionNum;
  UINT32            ParamVersionNum;
  DE_SETTING_s      DeInfo[24];
```

```
      TONE_CURVE_s      ToneCurve[6];
      UINT16            Vignette[6][1089];
    } DE_PARAM_s;
```

2.2.9 Data Structures: CALIBRATION_PARAM_s

This structure is defined in Section 2.3.8.1.

```
     typedef struct _CALIBRATION_PARAM_s_ {
      UINT32            VersionNum;
      UINT32            ParamVersionNum;
      UINT32            ParamVersionNum;
      UINT32            ParamVersionNum;
      UINT32            ParamVersionNum;
      VIG_LUMA_TABLE_s  VigLumaTable[4];
      WB_BLEND_CURVE_s  VigWbBlendTable[4];
     } CALIBRATION_PARAM_s;
```

2.3 IQ Tuning: Header File Details

The header files for IQ tuning are defined below:

- (Section 2.3.1) Header File Details: AmbaIQParamXXX_A12_DefaultParams.c
- (Section 2.3.2) Header File Details: AmbaIQParamXXX_A12_ADJ_VideoXXX.c
- (Section 2.3.3) Header File Details: AmbaIQParamXXX_A12_ADJ_StillLIso.c
- (Section 2.3.4) Header File Details: AmbaIQParamXXX_A12_ADJ_StillHIso.c
- (Section 2.3.5) Header File Details: AmbaIQParamXXX_A12_ImageParam.c
- (Section 2.3.6) Header File Details: AmbaIQParamXXX_A12_ScXXXParam.c
- (Section 2.3.7) Header File Details: AmbaIQParamXXX_A12_DeXXXParam.c

2.3.1 Header File Details: AmbaIQParamXXX_A12_DefaultParams.c

The structures in the header file `AmbaIQParamXXX_A12_DefaultParams.c` are as shown below. For more details related to the tuning parameters, refer to `AmbaImg_AaaDef.h`.

2.3.1.1 AmbaIQParamXXX_DefaultParams.c: Programming Map

- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AMBA_3A_OP_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_CONTROL_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_EV_LUT_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_EV_LUT_s > AE_EV_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s >AE_EV_LUT_s > AE_EV_INFO_s > MAX_DB_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_ISO_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_DEF_SETTING_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_DEF_SETTING_s > Video/StillFaceDetect
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_CONTROL_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > AWB_LUT_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > DEFAULTMENU_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > AWB_LUT_s >AWB_LUT_UINT_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s >FLASH_AE_AWB_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > EIS_CONTROL_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SC_LIGHT_CONDITION_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SC_LIGHT_CONDITION_s > HISTO_INFO_s
- AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SCENE_DETECT_CONDITION_s

2.3.1.1.1 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s

```
typedef struct _AAA_PARAM_s_ {
  UINT32      StructVersionNum;
  UINT32      ParamVersionNum;
  AMBA_3A_OP_INFO_s AaaFunc;
  AE_CONTROL_s      AeControlMode; AE_EV_LUT_s
  AeEvLut; AE_ALGO_INFO_s AeAlgoInfo;
  AWB_CONTROL_s     AwbControlMode;
  AWB_ALGO_INFO_s   AwbAlgoInfo;
  FLASH_AE_AWB_INFO_s FlashAeAwbInfo;
  AE_SENSOR_COMPEN_s      SensorCompenInfo;
  AF_CONTROL_s      AfControlMode;
  EIS_CONTROL_s     EisControlMode; SCENEMODE_s   SceneMode;
} AAA_PARAM_s;
```

The following table shows the fields of each parameter in the **AAA_PARAM_s** structure. The field definitions are given below:

Type	Field	Description
UINT32	**StructVersionNum**	Version number of this structure
UINT32	**ParamVersionNum**	Version number of these tuning parameters

structure	**AaaFunc**	AAA functions include the following:
		AE, AWB and AF function.
		AE => Auto Exposure
		AWB => Auto White Balance
		AF => Auto Focus
		
		Please refer to **AMBA_3A_OP_INFO_s** in Section 2.3.1.1.1.1.

structure	**AeControlMode**	Auto exposure control Please refer to **AE_CONTROL_s** in Section 2.3.1.1.1.2.
structure	**AeEvLut**	Auto exposure LUT Please refer to **AE_EV_LUT_s** in Section 2.3.1.1.1.3.
structure	**AeAlgoInfo**	Auto exposure algorithm Please refer to **AE_ALGO_INFO_s** in Section 2.3.1.1.1.6.
structure	**AwbControlMode**	Auto white balance control Please refer to **AWB_CONTROL_s** in Section 2.3.1.1.1.10.
structure	**AwbAlgoInfo**	Auto white balance algorithm Please refer to **AWB_ALGO_INFO_s** in Section 2.3.1.1.1.11.
structure	**FlashAeAwbInfo**	Flash auto exposure / white balance information Please refer to **FLASH_AE_AWB_INFO_s** in Section 2.3.1.1.1.15.
structure	**SensorCompenInfo**	TBD.
structure	**AfControlMode**	TBD.
structure	**EisControlMode**	Electronic image stabilization control Please refer to **EIS_CONTROL_s** in Section 2.3.1.1.1.16.
structure	**SceneMode**	Scene mode Please refer to **SCENEMODE_s** in Section 2.3.1.1.1.17.

*Table 2-1. Parameter definition for **AAA_PARAM_s()**.*

2.3.1.1.1.1 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AMBA_3A_OP_INFO_s

The definitions for **AMBA_3A_OP_INFO_s** are given in the following structure:

```
typedef struct _AMBA_3A_OP_INFO_s_ {
  UINT8 AeOp;
  UINT8 AwbOp;
  UINT8 AfOp;
  UINT8 AdjOp;
  UINT8 Reserved;
  UINT8 Reserved1;
  UINT8 Reserved2;
  UINT8 Reserved3;
} AMBA_3A_OP_INFO_s;
```

The field definitions are as follows:

Type	Field	Description
UINT8	**AeOp**	0: AE Off 1: AE On
UINT8	**AwbOp**	0: AWB Off 1: AWB On
UINT8	**AfOp**	0: AF Off 1: AF On
UINT8	**AdjOp**	0: Adjusting Function Off 1: Adjusting Function On

*Table 2-2. Field definition for **AAA_PARAM_s** structure **AMBA_3A_OP_INFO_s()**.*

2.3.1.1.1.2 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_CONTROL_s

```
typedef struct _AE_CONTROL_s_ {
    UINT16    DefAeTarget;
    UINT8     VideoManualModeEnable;
    UINT8     SlowShutter;
    UINT8     SlowShutterFps;     //60fps,50fps, 30fps,25fps, 15fps,12fps,
                                7fps,6fps
    UINT8     PhotoManualModeEnable;
    UINT8     PhotoSlowShutter;
    UINT8     PhotoSlowShutterFps;//60fps,50fps, 30fps,25fps, 15fps,12fps,
                                7fps,6fps
    UINT8     VideoAeSpeed;       //0~6
    UINT8     PhotoAeSpeed;       //0~6
    UINT8     MeteringMode;
    UINT8     Backlight;
    INT16     EvBias;
    UINT8     VideoFlash;
    UINT8     TargetStatisticDiffEnable;
```

```
        UINT8     StillFlash;
        UINT8     StillFlashType;
        UINT8     StillFlashFlickerChk;   //0~4
        UINT8     StillNightShot;
        UINT8     StillIs;


        UINT8     StillFlickerChk;  //0~4
        UINT16    StillIris;
        UINT16    StillIso;
        UINT16    StillShutter;

        UINT8     StillPMode;
        INT16     StillPModeStr;       //1EV : 128

        UINT8     StillNormalPost;
        UINT8     StillFlashPost;
        UINT8     StillContinousPost;

        UINT8     TimeLapseEnable;
        UINT32    TimeLapseSkipFrames;

        UINT8     DualMainVideoEnable;
        UINT8     DualSecondVideoEnable;
        UINT8     DualSecondVideoAeTarget;
        INT16     DualSecondVideoEvBias;
        UINT8     DualSecondVideoAeSpeed;
        UINT8     DualSecondVideoMeteringMode;

        INT32     StillContinueSpeed;       //0~128
    } AE_CONTROL_s;
```

The field definitions are as follows:

Type	Field	Description
UINT16	**DefAeTarget**	Default auto exposure (AE) target value. If set to 0, the affected AE target will be one of those specified in `AmbaIQParamXXX_A12_Adj_XXX.c`. The default AE target takes effect only when it has a non-zero value. (AE target values specified in AmbaIQ`ParamXXX_A12_Adj_XXX.c` will be ignored)
UINT8	**VideoManualModeEnable**	TBD.
UINT8	**SlowShutter**	DISABLE ENABLE (video mode)
UINT8	**SlowShutterFps**	Specifies the frames per second (FPS) number for video mode slow shutter. It can be: 60, 50, 30, 25, 15, 12, 7, 6 (fps).
UINT8	**PhotoManualModeEnable**	TBD.
UINT8	**PhotoSlowShutter**	DISABLE ENABLE (still mode)
UINT8	**PhotoSlowShutterFps**	Specifies the FPS number for still mode slow shutter, it can be: 60, 50, 30, 25, 15, 12, 7, 6 (fps).
UINT8	**VideoAeSpeed**	Specifies the video AE converging speed and the value is between 0 - 6. 0 is high converging speed and 6 is low converging speed.

Type	Field	Description
UINT8	**PhotoAeSpeed**	Specifies the photo-preview AE converging speed and the value is between 0 ~ 6. 0 is high-converging speed and 6 is low-converging speed.
UINT8	**MeteringMode**	**AE_METER_CENTER** **AE_METER_AVERAGE** **AE_METER_SPOT** In different metering modes, the spatial weights for integrated digital signal processing (IDSP) tile statistics will be different.
UINT8	**Backlight**	TBD.
INT16	**EvBias**	Bias for the AE target, used for increasing or decreasing the AE target. The effect values are -96 ~ 96. -96 => -3EV -64 => -2EV -32 => -1EV 0 => 0EV 32 => 1EV 64 => 2EV 96 => 3EV
UINT8	**VideoFlash**	**FLASH_ALWAYS_OFF** **FLASH_AUTO** **FLASH_ALWAYS_ON** Turn off/on flash for video mode.
UINT8	**TargetStatisticDiffEnable**	TBD.
UINT8	**StillFlash**	**FLASH_ALWAYS_OFF** **FLASH_AUTO** **FLASH_ALWAYS_ON** Turn off/on flash for still mode.
UINT8	**StillFlashType**	TBD.
UINT8	**StillFlashFlickerChk**	In order to prevent the flicker defect in the flash mode, this parameter is used to modify the shutter time. The value should be 0 - 4. 0: No check 1: 1/120 - 1/60 second 2: 1/120 - 1/30 second 3: 1/120 - 1/15 second 4: 1/120 - 1/7.5 second If the shutter speed is estimated to be between the selected shutter range, the AE will change the shutter speed to prevent the flicker defect according to the current flicker mode.
UINT8	**StillNightShot**	DISABLE ENABLE If disabled, the shutter value can be up to **still_shutter_min_normal**. If enabled, the shutter value can be up to **still_shutter_min_night**.
UINT8	**StillIs**	DISABLE ENABLE Turn off/on Image Stabilization (IS) for still mode.

Type	Field	Description
UINT8	**StillFlickerChk**	In order to prevent the flicker defect, this parameter is used to modify the shutter time. The value should be 0 - 4. 0: No check 1: 1/120 - 1/60 second 2: 1/120 - 1/30 second 3: 1/120 - 1/15 second 4: 1/120 - 1/7.5 second If the shutter speed is estimated between the selected shutter range, the AE will change the shutter speed to prevent the flicker defect according to the current flicker mode.
UINT16	**StillIris**	**AE_IRIS_AUTO** **AE_IRIS_FIX** If this parameter is not set to be **AE_IRIS_AUTO**, it should be set to be a correct iris index. The iris index is decided by the lens driver.
UINT16	**StillIso**	**AE_ISO_AUTO** **AE_ISO_AUTO_HISO** **AE_ISO_3** **AE_ISO_6** **AE_ISO_12** **AE_ISO_25** **AE_ISO_50** **AE_ISO_100** **AE_ISO_200** **AE_ISO_400** **AE_ISO_800** **AE_ISO_1600** **AE_ISO_3200** **AE_ISO_6400** **AE_ISO_12800** **AE_ISO_25600** **AE_ISO_51200** **AE_ISO_102400** **AE_ISO_204800** This parameter is used to set the ISO mode.
UINT16	**StillShutter**	**AE_SHUTTER_AUTO** Other Shutter Index If this parameter is not set to be **AE_SHUTTER_AUTO**, it should be set to be a correct shutter index. Shutter index is decided by the sensor driver.
UINT8	**StillPMode**	**AE_P_MODE_OFF** **AE_P_MODE_ISO_IRIS** **AE_P_MODE_ISO_SHUTTER** **AE_P_MODE_IRIS_ISO** **AE_P_MODE_IRIS_SHUTTER** **AE_P_MODE_SHUTTER_ISO** **AE_P_MODE_SHUTTER_IRIS** This parameter is used to set the **P_MODE**(programmable still exposure control).

Type	Field	Description
INT16	**StillPModeStr**	This parameter is used to set the strength of the **P_MODE** control. The value = 128 implies 1 EV, and the value = -128 implies -1 EV. If the StillPMode is set to be **AE_P_ MODE_OFF**, the still exposure will be estimated automatically. If not, the exposure value will be modified according the value (estimated with the StillPMode = AE_P_MODE_OFF) and the p_mode setting. For example, if the estimated exposure value (estimated with the StillPMode = AE_P_MODE_OFF) is shutter = 1/30; then, ISO = 200; F-number = F4 When the StillPMode = AE_P_MODE_ISO_IRIS and the StillPModeStr = 128, the exposure value will be shutter = 1/30; ISO = 400; F-number = F5.6. When the StillPMode = AE_P_MODE_ISO_SHUTTER and the StillPModeStr = 128, the exposure value will be shutter = 1/60; ISO = 400; F-number = F4. When the StillPMode = AE_P_MODE_IRIS_ISO and the StillPModeStr = 128, the exposure value will be shutter = 1/30; ISO = 100; F-number = F2.8. When the StillPMode = AE_P_MODE_IRIS_SHUTTER and the StillPModeStr = 128, the exposure value will be shutter = 1/60; ISO = 200; F-number = F2.8. When the StillPMode = AE_P_MODE_SHUTTER_ISO and the StillPModeStr = 128, the exposure value will be shutter = 1/15; ISO = 100; F-number = F4. When the StillPMode = AE_P_MODE_SHUTTER_IRIS and the StillPModeStr = 128, the exposure value will be shutter = 1/15; ISO = 200; F-number = F5.6.
UINT8	**StillNormalPost**	DISABLE ENABLE This parameter is used to turn on/off the Still AE post processing.
UINT8	**StillFlashPost**	DISABLE ENABLE This parameter is used to turn on/off the Still Flash AE post processing.
UINT8	**StillContinousPost**	TBD.
UINT8	**TimeLapseEnable**	DISABLE ENABLE This parameter is used to turn on/off the AE lapse function.
UINT32	**TimeLapseSkipFrames**	This parameter is used to set the skip frames for the AE lapse function. The AE lapse function is used to set the frames between the two successive AE processes.
UINT8	**DualMainVideoEnable**	DISABLE ENABLE This parameter is used to turn on/off the AE for the main video.

Type	Field	Description
UINT8	**DualSecondVideoEnable**	DISABLE ENABLE These parameters are used to turn on/off the AE for the second video.
UINT8	**DualSecondVideoAeTarget**	Specifies the AE target for the second video.
INT16	**DualSecondVideoEvBias**	Specifies the exposure value (EV) bias for the second video.
UINT8	**DualSecondVideoAeSpeed**	Specifies the second video AE converging speed and the value is between 0 - 6. 0 is high-converging speed, and 6 is low-converging speed.
UINT8	**DualSecondVideoMeteringMode**	Specifies the metering mode for the second video **AE_METER_CENTER** **AE_METER_AVERAGE** and **AE_METER_SPOT**.
UINT8	**StillContinousPost**	TBD.
UINT8	**TimeLapseEnable**	DISABLE ENABLE This parameter is used to turn on/off the AE lapse function.
UINT32	**TimeLapseSkipFrames**	This parameter is used to set the skip frames for the AE lapse function. The AE lapse function is used to set the frames between the two successive AE processes.
UINT8	**DualMainVideoEnable**	DISABLE ENABLE This parameter is used to turn on/off the AE for the main video.
UINT8	**DualSecondVideoEnable**	DISABLE ENABLE This parameters is used to turn on/off the AE for the second video.
UINT8	**DualSecondVideoAeTarget**	Specifies the AE target for the second video.
INT16	**DualSecondVideoEvBias**	Specifies the EV bias for the second video.
UINT8	**DualSecondVideoAeSpeed**	Specifies the second video AE converging speed and the value is between 0 - 6. 0 is high-converging speed and 6 is low-converging speed.
UINT8	**DualSecondVideoMetering**	Specifies the metering mode for the second video **AE_METER_CENTER** **AE_METER_AVERAGE** and **AE_METER_SPOT**.
INT32	**StillContinueSpeed**	0 ~ 128 0: The fastest converging speed 128: The slowest converging speed This parameter is used to set the AE converge speed in the continuous still capture mode.

*Table 2-3.   Field definition for the **AAA_PARAM_s** structure **AE_CONTROL_s()**.*

2.3.1.1.1.3 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_EV_LUT_s

The definitions for **AE_EV_LUT_s** are:

```
typedef struct _AE_EV_LUT_s_ {
  UINT8  FlickerMode;
  UINT8  HalfFrameShutter;
  UINT16 VideoShutterMaxIdx;
  AE_EV_INFO_s      Lut;
  UINT8  VinTotal;
  UINT8  TilesSide; // 0:All    1:Left      2:Right
} AE_EV_LUT_s;
```

The field definitions for **AE_EV_LUT_s** are as follows:

Type	Field	Description
UINT8	**FlickerMode**	**ANTI_FLICKER_AUTO** **ANTI_FLICKER_60HZ** **ANTI_FLICKER_50HZ** Specifies the power frequency for anti-flicker or check automatically.
UINT8	**HalfFrameShutter**	DISABLE ENABLE Enabling HalfFrameShutter shortens the shutter time. If the current minimum shutter is 1/30 second, this allows the shutter time to be reduced to 1/60 second.
UINT16	**VideoShutterMaxIdx**	Specifies the maximum shutter (index) for the video mode.
structure	**Lut**	Please refer to **AE_EV_INFO_s** definition in Section 2.3.1.1.1.4.
UINT8	**VinTotal**	TBD
UINT8	**TilesSide**	TBD

*Table 2-4. Field Definition for the **AAA_PARAM_s** Structure **AE_EV_LUT_s()**.*

2.3.1.1.1.4 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_EV_LUT_s > AE_EV_INFO_s

The definitions for **AE_EV_INFO_s** are as follows:

```
typedef struct _AE_EV_INFO_s_ {
  INT16 VideoMinAgcInfo;  //0db~6db
  INT16 StillMinAgcInfo;  //0db~6db
  MAX_DB_INFO_s     MaxAgcInfo[6];    //9db~27db
  MAX_DB_INFO_s     MaxDGainInfo[6];  //9db~27db
} AE_EV_INFO_s;
```

The field definitions AE_EV_INFO_s are:

Type	Field	Description
INT16	**VideoMinAgcInfo**	Specifies the minimum AGC (dB) for the video mode.
INT16	**StillMinAgcInfo**	Specifies the minimum AGC (dB) for the still mode.
structure	**MaxAgcInfo[6]**	Please refer to **MAX_DB_INFO_s** definition in Section 2.3.1.1.1.5.
structure	**MaxDGainInfo[6]**	Please refer to **MAX_DB_INFO_s** definition in Section 2.3.1.1.1.5.

*Table 2-5. Field Definition for the **AAA_PARAM_s** structure **AE_EV_INFO_s()**.*

2.3.1.1.1.5 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s >AE_EV_LUT_s > AE_EV_INFO_s > MAX_DB_INFO_s

This structure defines the max D-Gains/AGCs of 60/30/15/7.5/240/120 fps for each **max_agc_info** [0 to 5] (non binning/binning_2x/binning_4x/binning_8x/photo_mode/NA, 6 total).

The definitions for **MAX_DB_INFO_s** are as follows:

```
typedef struct MAX_DB_INFO_s_ {
  UINT16      Db60Fps;
  UINT16      Db30Fps;
  UINT16      Db15Fps;
  UINT16      Db7p5Fps;
  UINT16      Db240Fps;
  UINT16      Db120Fps;
} MAX_DB_INFO_s;
```

The field definitions for **MAX_DB_INFO_s** are as follows:

Type	Field	Description
UINT16	**Db60Fps**	Maximum gain for 60fps
UINT16	**Db30Fps**	Maximum gain for 30fps
UINT16	**Db15Fps**	Maximum gain for 15fps
UINT16	**Db7p5Fps**	Maximum gain for 7.5fps
UINT16	**Db240Fps**	Maximum gain for 240fps
UINT16	**Db120Fps**	Maximum gain for 120fps

*Table 2-6. Field Definition for the **AAA_PARAM_s** Structure **MAX_DB_INFO_s()**.*

2.3.1.1.1.6 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s

The definitions for **AE_ALGO_INFO_s** are as follows:

```
typedef struct _AE_ALGO_INFO_s_ {
AE_ISO_INFO_s AeVideoIsoInfo;
AE_ISO_INFO_s AeStillIsoInfo;
AE_DEF_SETTING_s    DefSetting;
UINT8               RoiCnt;
UINT8               RoiInfo[10][96];
} AE_ALGO_INFO_s;
```

The field definitions for **AE_ALGO_INFO_s** are as follows:

Type	Field	Description
structure	**AeVideoIsoInfo**	Please refer to **AE_ISO_INFO_s** in Section 2.3.1.1.1.7.
structure	**AeStillIsoInfo**	
structure	**DefSetting**	Please refer to **AE_ISO_INFO_s** in Section 2.3.1.1.1.8.
UINT16	**RoiCnt**	Specifies the number of weights in the AE ROI; usually set as 96. (8 x 12).
UINT8	**RoiInfo[10][AE_MAX_TILES]**	Specifies the weights of each tile in the ROI for every metering method. **AE_MAX_TILES** is the number of tiles in the AE ROI (typically 96). These values are set for the AE weighting tables. Currently, there are three kinds of AE-Metering modes: **AE_METER_CENTER**, **AE_METER_AVERAGE**, and **AE_METER_SPOT**.

*Table 2-7. Field definition for the **AAA_PARAM_s** structure **AE_ALGO_INFO_s()**.*

2.3.1.1.1.7 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_ISO_INFO_s

The structure **AE_ISO_INFO_s** defines the video/still mode AGC index for each ISO control. The index is used to calculate the ISO value embedded in the Exchange image file format (EXIF) information. The value “2048” is the index of the **video_min_agc/still_ min_agc**, and the step is 128 for 1-EV. For example, if 2048 was used to map 0db (ISO100), the value “2176” means -6db (ISO50) and the value “1920” means 6db(ISO200).

```
typedef struct _AE_ISO_INFO_s_ {
  INT32 Iso3AgcIdx;
  INT32 Iso6AgcIdx;
  INT32 Iso12AgcIdx;
  INT32 Iso25AgcIdx;
  INT32 Iso50AgcIdx;
  INT32 Iso100AgcIdx; // min_agc      db
  INT32 Iso200AgcIdx; // min_agc+     6 db
  INT32 Iso400AgcIdx; // min_agc+     12 db
  INT32 Iso800AgcIdx; // min_agc+     18 db
  INT32 Iso1600AgcIdx; // min_agc+    24 db
  INT32 Iso3200AgcIdx; // min_agc+    30 db
  INT32 Iso6400AgcIdx; // min_agc+    36 db
  INT32 Iso12800AgcIdx; // min_agc+   42 db
```

```
    INT32 Iso25600AgcIdx; // min_agc+   48 db
    INT32 Iso51200AgcIdx; // min_agc+   56 db
    INT32 Iso102400AgcIdx; // min_agc+  64 db
    INT32 Iso204800AgcIdx; // min_agc+  72 db
  } AE_ISO_INFO_s;
```

The field definitions are as follows:

Type	Field	Description
INT32	**Iso3AgcIdx**	Reference AGC Index for ISO3
INT32	**Iso6AgcIdx**	Reference AGC Index for ISO6
INT32	**Iso12AgcIdx**	Reference AGC Index for ISO12
INT32	**Iso25AgcIdx**	Reference AGC Index for ISO25
INT32	**Iso50AgcIdx**	Reference AGC Index for ISO50
INT32	**Iso100AgcIdx**	Reference AGC Index for ISO100
INT32	**Iso200AgcIdx**	Reference AGC Index for ISO200
INT32	**Iso400AgcIdx**	Reference AGC Index for ISO400
INT32	**Iso800AgcIdx**	Reference AGC Index for ISO800
INT32	**Iso1600AgcIdx**	Reference AGC Index for ISO1600
INT32	**Iso3200AgcIdx**	Reference AGC Index for ISO3200
INT32	**Iso6400AgcIdx**	Reference AGC Index for ISO6400
INT32	**Iso12800AgcIdx**	Reference AGC Index for ISO12800
INT32	**Iso25600AgcIdx**	Reference AGC Index for ISO25600
INT32	**Iso51200AgcIdx**	Reference AGC Index for ISO51200
INT32	**Iso102400AgcIdx**	Reference AGC Index for ISO102400
INT32	**Iso204800AgcIdx**	Reference AGC Index for ISO204800

*Table 2-8.   Field Definition for the **AAA_PARAM_s** Structure **AE_ISO_INFO_s()**.*

2.3.1.1.1.8 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_DEF_SETTING_s

The definitions for **AE_DEF_SETTING_s** are as follows:

```
  typedef struct _AE_DEF_SETTING_s_ {
    UINT32      GlobalDGain;
    UINT16      VideoDarkLuma;
    UINT16      PhotoPreviewDarkLuma;
    UINT16      StillDarkLuma;

    UINT16      StillShutterMinNormal;
    UINT16      StillShutterMinNight;
    UINT16      StillShutterMinFlash;
    UINT16      StillShutterMinFlashSlow;
    UINT16      StillShutterMinContinue;
    UINT16      StillShutterMinBurst;
    UINT16      StillShutterMaxNormal;
    UINT16      StillShutterMaxFlash;
    AE_FACE_DETECTION_s     VideoFaceDetect;
    AE_FACE_DETECTION_s     StillFaceDetect;

    UINT8 HighLightLvNo;

    UINT8 OutdoorLvNo;
    UINT8 LowLightLvNo;
    LUT_CONTROL_s     FlashFocusDistanceTargetRatio; //unit:128
  } AE_DEF_SETTING_s;
```

The field definitions for **AE_DEF_SETTING_s** are as follows:

Type	Parameter	Description
UINT16	**GlobalDGain**	Specifies the global D-gain value. 4096 is the unit gain.
UINT16	**VideoDarkLuma**	Specifies the dark luma values for video, photo preview, and still mode. The dark luma value is used to suppress noise in very-low light conditions. When the AGC gain reaches the maximum and the current luma statistic from IDSP is still lower than **dark_luma**, the automatic gain control (AGC) AGC gain will decrease to make the noise invisible. The suggested dark luma value is 10.
UINT16	**PhotoPreviewDarkLuma**	
UINT16	**StillDarkLuma**	
UINT16	**StillShutterMinNormal**	Defines the minimum-shutter index for the normal mode.
UINT16	**StillShutterMinNight**	Defines the minimum-shutter index for the night mode (takes effect if **still_night_shot** is set to enable).
UINT16	**StillShutterMinFlash**	Defines the minimum-shutter index for the normal mode.
UINT16	**StillShutterMinFlashSlow**	Defines the minimum-shutter index for the flash slow mode.
UINT16	**StillShutterMinContinue**	Defines the minimum-shutter index for the continue mode.
UINT16	**StillShutterMinBurst**	Defines the minimum-shutter index for the burst mode.
UINT16	**StillShutterMaxNormal**	Defines the maximum-shutter index for normal mode.
UINT16	**StillShutterMaxFlash**	Defines the maximum-shutter index for flash mode. The shutter indexes are used to limit the longest and the shortest exposure time for the various modes. Please refer to the sensor driver for a suitable value.
structure	**VideoFaceDetect**	Please refer to **AE_FACE_DETECTION_s** in Section 2.3.1.1.1.9.
structure	**StillFaceDetect**	
UINT8	**HighLightLvNo**	Specifies the lowest Luma value (LV) of a high light environment.
UINT8	**OutdoorLvNo**	Specifies the lowest LV of an outdoor environment.
UINT8	**LowLightLvNo**	Specifies the highest LV of a low light environment.
structure	**FlashFocusDistanceTargetRatio**	Specifies AE target ratio per different focus distance (Unit 128).

Table 2-9.   Field definition for the ***AAA_PARAM_s*** *structure* ***AE_DEF_SETTING_s()****.*

2.3.1.1.1.9 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AE_ALGO_INFO_s > AE_DEF_SETTING_s > Video/StillFaceDetect

The definitions for **AE_FACE_DETECTION_s** are:

```
typedef struct _AE_FACE_DETECTION_s_ {
  UINT8 Enable;                 //EnableAeFaceDetection;
  UINT8 TargetGain;             //FaceDetection_exposure_Target_Gain to
                                //normal ae Target, unit:128
  UINT8 LumaPriority;           //FaceLumaPriority 0 ~ 16
  UINT8 GetFaceDelayFps;        //get_FacedelayFps relate to 30Fps
  UINT8 AfterGetFaceDelayFps;   //after_get_FacedelayFps relate to 30Fps
} AE_FACE_DETECTION_s;
```

The field definitions for **AE_FACE_DETECTION_s** are as follows:

Type	Field	Description
UINT8	**Enable**	0: DISABLE 1: ENABLE (face detection)
UINT8	**TargetGain**	When face detection is enabled, this parameter will be used to generate the **Face_AE_Target** value. 128 is the unit value. Face_AE_Target = Normal_AE_Target * target_gain / 128.
UINT8	**LumaPriority**	When face detection is enabled, this parameter is used to generate the **Final_AE_Target** value, and the range of the value is 0 - 16. Final_AE_Target = (Normal_AE_target * (16 – luma_priority) + Face_AE_Target * luma_priority) / 16.
UINT8	**GetFaceDelayFps**	This specifies the number of delay frames before applying the Final_AE_Target value described above. The unit for **delay_fps** is 30fps. If the **GetFaceDelayFps** value is 15, the **Final_AE_Target** value will be applied after 15/30 of a second once the face is detected.
UINT8	**AfterGetFaceDelayFps**	This specifies the number of delay frames before canceling the Final_AE_Target value described above. The unit of delay_fps is 30fps. If the **AfterGetFaceDelayFps** value is 15, the **Final_AE_Target** value will be disabled after 15/30 of a second once the face is no longer detected.

*Table 2-10. Field Definition for the **AAA_PARAM_s** Structure **AE_FACE_DETECTION_s()**.*

2.3.1.1.1.10 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_CONTROL_s

The definitions for **AWB_CONTROL_s** are:

```
typedef struct _AWB_CONTROL_s_ {
  UINT8 Method;
  UINT8 Speed;
  UINT8 VideoSkipFrames;
  UINT8 PhotoSkipFrames;
  Uint8 MenuMode;
  Uint8 MenuModeType;
  Uint8 MeteringMode;
  Uint8 VideoNoWhite;
  Uint8 StillNoWhite;

  INT32 StillContinueSpeed;

  UINT8 TileSideChk[5];
} AWB_CONTROL_s;
```

The field definitions for **AWB_CONTROL_s** are:

Type	Field	Description
UINT8	**Method**	Specifies the automatic white balance (AWB) method 1: Gray world 2: White patch (suggested)
UINT8	**Speed**	1 - 64: Manual 128: Auto The AWB algorithm will update the AWB gain in the video mode. This parameter controls the update speed. If the speed is 128, the update speed will be controlled by the original setting in the AWB algorithm. If the speed is 1 - 64, the update speed will be controlled by the following formula: AWB_next_gain = (estimated_gain * speed + current_gain * (64 – speed)) / 64
UINT8	**VideoSkipFrames**	Skips FPS for AWB in video mode. If the frame rate is 30 fps and the VideoSkipFrames parameter is 29, the AWB gain in video mode will be calculated and updated once every 30 frames.
UINT8	**PhotoSkipFrames**	Skips FPS for AWB in still mode. If the frame rate is 30 fps and the **still_skip_frames** parameter is 4, the AWB gain in the still mode will be calculated and updated once every 5 frames.
UINT8	**MenuMode**	This index is used to set the AWB mode. Settings include **WB_AUTOMATIC**, **INCANDESCENT**, **SUNNY**, etc. Please refer to the related file.
UINT8	**MenuModeType**	Specifies the method of generating the AWB gain when the user enters the menu mode. **WB_MENU_REGION**: The AWB gain is generated by referring to the specified white region according to the user ’s selection in the white balance (WB) menu. **WB_MENU_FIX**: The AWB gain is set directly to the default gain according to the user’s selection in the WB menu.

Type	Field	Description
UINT8	**MeteringMode**	Specifies the metering method for AWB. **AWB_METER_AVERAGE** **AWB_METER_CENTER** **AWB_METER_SPOT**.
UINT8	**VideoNoWhite**	**AWB_NO_WHITE_LAST_GAIN** **AWB_NO_WHITE_DEFAULT_GAIN** This parameter is used to set the WB-gain when no white video is detected by the AWB algorithm. When no white video is detected by **AWB algorithm**, the **AWB_NO_WHITE_LAST_GAIN** will apply the last detected WB-gain and the **AWB_NO_ WHITE_DEFAULT_GAIN** will apply the default WB-gain.
UINT8	**StillNoWhite**	0 ~ 128 0: The fastest converge speed 128: The slowest converge speed This parameter is used to set the AWB converge speed in the continuous still capture mode.
UINT8	**StillContinueSpeed**	
UINT8	**TileSideChk[5]**	TBD

*Table 2-11. Field Definition for the **AAA_PARAM_s** Structure **AWB_CONTROL_s()**.*

2.3.1.1.1.11 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s

The definitions for **AWB_ALGO_INFO_s** are:

```
typedef struct _AWB_ALGO_INFO_s_ {
  AWB_LUT_s         WhiteRegionLut;
  UINT8             AwbLutNum[20][2];
  DEFAULTMENU_s     DefMenuInfo[20];

  UINT16            RoiCnt;
  UINT8             RoiInfo[10][WB_MAX_TILES];
} AWB_ALGO_INFO_s;
```

The field definitions for AWB_ALGO_INFO_s are:

Type	Field	Description
structure	**WhiteRegionLut**	Please refer to **AWB_LUT_s** in Section 2.3.1.1.1.12 below.
UINT8	**AwbLutNum[20][2]**	These parameters are used to set the white regions according to the white balance (WB) mode. The first index is set for the white region start, and the second index is set for the white region number. For instance, if the setting is {2, 2}, //LUT num. SUNNY, the white region will be the OR function (it is a logic function, like “0 OR 0 is 0”, “0” OR “1” is “1”...) (OR) of the second white region and the third white region in the Sunny WB mode.
structure	**DefMenuInfo[20]**	Please refer to **DEFAULTMENU_s** in Section 2.3.1.1.1.13 below.
UINT16	**RoiCnt**	Specifies the number of weights in the AWB Region of interested (ROI).
UINT8	**RoiInfo[10][WB_MAX_TILES]**	Specifies the weighting of each tile in the AWB ROI for every metering method. **WB_MAX_TILES** is the number of tiles in the AWB ROI.

*Table 2-12. Field definition for the **AAA_PARAM_s** structure **AWB_ALGO_INFO_s()**.*

2.3.1.1.1.12 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > AWB_LUT_s

The definitions for **AWB_LUT_s** are:

```
typedef struct _AWB_LUT_s_ {
  UINT8                Lut_No;
  AWB_LUT_UINT_s       AwbLut[20];
  UINT8                LumaWg[64];
} AWB_LUT_s;
```

The field definitions are:

Type	Field	Description
UINT8	**Lut_No**	Specifies the total number of the white regions for various environments. The legal range is 1 - 20.
structure	**AwbLut[20]**	Please refer to **AWB_LUT_UINT_s** in Section 2.3.1.1.1.14 below.
UINT8	**LumaWg[64]**	This parameter is used to set the luma-weight value for the AWB algorithm. The entry numbers are 64, and this is linearly mapped to the range 0 - 63.

*Table 2-13. Field Definition for the **AAA_PARAM_s** structure **AWB_LUT_s()**.*

2.3.1.1.1.13 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > DEFAULTMENU_s

The structure **DEFAULTMENU_s** is defined as follows:

```
typedef struct _DEFAULTMENU_s_ {
  AMBA_DSP_IMG_WB_GAIN_s  WbGain;
  UINT16               RgbRatio[3];
} DEFAULTMENU_s;
```

The field definitions for **DEFAULTMENU_s** are:

Type	Field	Description
structure	**AMBA_DSP_IMG_WB_GAIN_s**	These variables are used to set the default WB gains according to the WB mode, and the unit value is 4096. If the AWB gain cannot be obtained, these default WB gains are used.
UINT16	**RgbRatio[3]**	Specifies the unit gain for R, G, and B components in each WB mode.

*Table 2-14. Field Definition for the **AAA_PARAM_s** Structure **DEFAULTMENU_s()**.*

2.3.1.1.1.14 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > AWB_ALGO_INFO_s > AWB_LUT_s >AWB_LUT_UINT_s

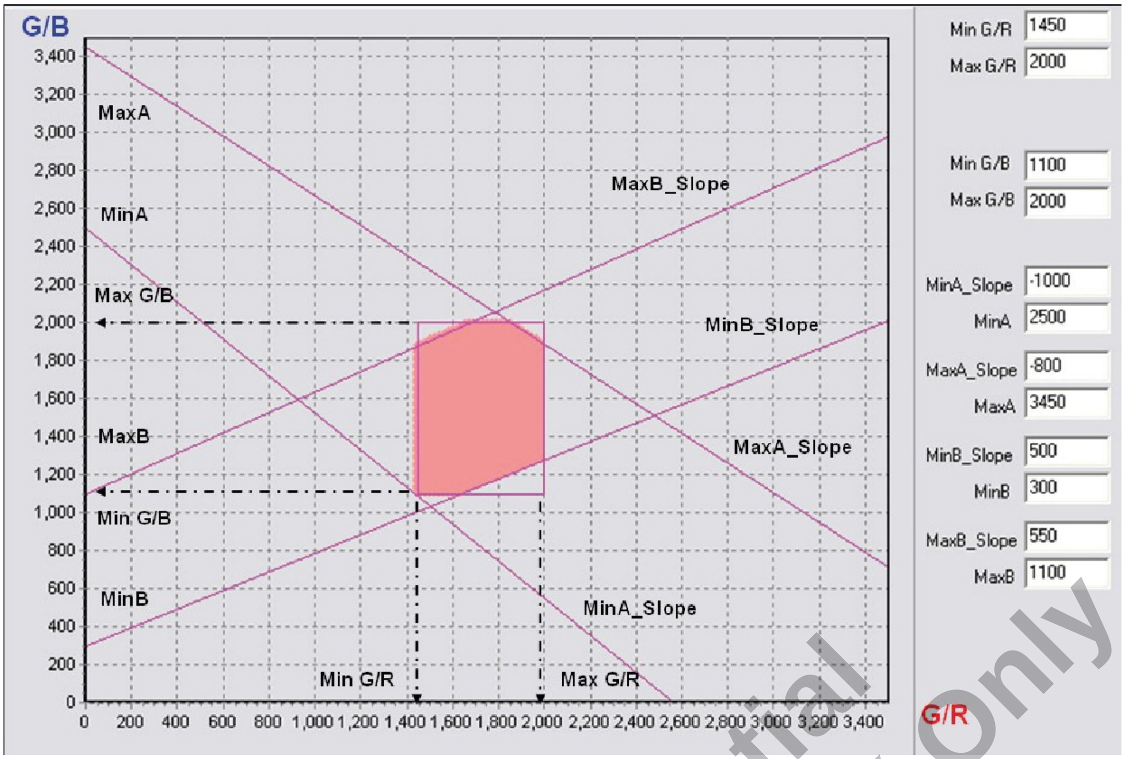
The structure **AWB_LUT_UINT_s** is defined as follows:

```
typedef struct _AWB_LUT_UINT_s_ {
  INT32 GrMin;
  INT32 GrMax;
  INT32 GbMin;
  INT32 GbMax;
  INT32 YAMinSlope;
  INT32 YAMin;
  INT32 YAMaxSlope;
  INT32 YAMax;
  INT32 YBMinSlope;
  INT32 YBMin;
  INT32 YBMaxSlope;
  INT32 YBMax;
  INT32 Weight;
} AWB_LUT_UINT_s;
```

The field definitions for **AWB_LUT_UINT_s** are:

Type	Field	Description
UINT16	**GrMin**	These fields define the 2D boundary for a white region in R ratio-B ratio plane. The unit values are 4096. In the structure **AWB_LUT_UINT_s**, the weight is used to set the AWB check by AWB Algorithm of Ambarella (Please refer to INT8 below).
UINT16	**GrMax**	
UINT16	**GbMin**	
UINT16	**GbMax**	
UINT16	**YAMinSlope**	
UINT16	**YAMin**	
UINT16	**YAMaxSlope**	
UINT16	**YAMax**	
UINT16	**YBMinSlope**	
UINT16	**YBMin**	
UINT16	**YBMaxSlope**	
UINT16	**YBMax**	
INT8	**Weight**	-1: If the AWB algorithm detects an outdoor condition and the white point is located in this region, then the white point will be truncated. Please refer to Figure 2-1 for the meaning of the AWB white region. 0: Skip this white region (not checked) > 0: This region will take effect with the weighting value, and it depends on the AWB modes, including Auto, INCANDESCENT, Sunny and so on.

*Table 2-15. Field Definition for the **AAA_PARAM_s** Structure **AWB_LUT_UINT_s()**.*



Min G/R 1450
Max G/R 2000
Min G/B 1100
Max G/B 2000
MinA_Slope -1000
MinA 2500
MaxA_Slope -800
MaxA 3450
MinB_Slope 500
MinB 300
MaxB_Slope 550
MaxB 1100

Figure 2-1. AWB White Region.

2.3.1.1.1.15 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s >FLASH_AE_AWB_INFO_s

The definitions for **FLASH_AE_AWB_INFO_s** are:

```
typedef struct _FLASH_AE_AWB_INFO_s {
  UINT16                  VideoFlashAutoOnIso;
  UINT16                  VideoFlashAutoOnShutter;
  UINT16                  VideoFlashAutoOnIris;
  UINT16                  StillFlashAutoOnIso;
  UINT16                  StillFlashAutoOnShutter;
  UINT16                  StillFlashAutoOnIris;
  UINT16                  PreFlashTimes;
  UINT16                  PreFlashStrength;
  UINT16                  ChargeVsync;
  INT16                   ChkRows[20];
  INT16                   ChkCols[20];
  INT32                   PreFlashBaseCalibRation;
  LUT_CONTROL_s           ZoomVector;
  FLASH_PREFLASH_INFO_s   PreFlashInfo[5];
  INT32                   PreFlashBase;
  UINT16                  Reserved0;
  UINT16                  Reserved1;
  UINT16                  Reserved2;
  INT16                   Reserved3[20];
} FLASH_AE_AWB_INFO_s;
```

The field definitions for **FLASH_AE_AWB_INFO_s** are:

Type	Field	Description
UINT16	**VideoFlashAutoOnIso**	These parameters are used to set the auto-flash threshold. The parameters, ISO, shutter and iris, could be regarded as a exposure value. The exposure value “iso = 100, shutter = 1/30 sec and iris = F2.8” is equal to the exposure value “iso = 200, shutter = 1/60 sec and iris = F2.8”.
UINT16	**VideoFlashAutoOnShutter**	
UINT16	**VideoFlashAutoOnIris**	
UINT16	**StillFlashAutoOnIso**	
UINT16	**StillFlashAutoOnShutter**	
UINT16	**StillFlashAutoOnIris**	
UINT16	**PreFlashTimes**	Specifies the number of preflashes.
UINT16	**PreFlashStrength**	Sets the preflash strength.
UINT16	**ChargeVsync**	Specifies the flash charge time in terms of VSyncs. A higher value results in a shorter interval between flashes and weaker flash strength.
INT16	**ChkRows[20]**	Specifies which rows in AE statistics will be used to calculate preflash luma statistics.
INT16	**ChkCols[20]**	Specifies which columns in AE statistics will be used to calculate preflash luma statistics.
INT32	**PreFlashBaseCalibRation**	Specifies the preflash luma ratio.
structure	**ZoomVector**	Specifies the adaptation of the flash related information per different O.zoom/ D.zoom step.
structure	**PreFlashInfo[5]**	Specifies the luma statistics value and preflash strength per different distance.
INT32	**PreFlashBase**	Specifies the luma value of the golden set for a certain distance.

*Table 2-16. Field Definition for the **AAA_PARAM_s** structure **FLASH_AE_AWB_INFO_s()**.*

2.3.1.1.1.16 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > EIS_CONTROL_s

The definitions for **EIS_CONTROL_s** are:

```
typedef struct_EIS_CONTROL_s_ {
  UINT8 VideoEisSamplingRate;
  UINT8 StillEisSamplingRate;
  UINT8 MaxEisRangePerct;
  UINT8 MinEisRangePerct;
  UINT8 FlgEisDzEffect;
  UINT8 XEisStrength;
  UINT8 YEisStrength;
  UINT8 XRscStrength;
  UINT8 YRscStrength;
  UINT8 MinSupportFr;
} EIS_CONTROL_s;
```

The field definitions for **EIS_CONTROL_s** are:

Type	Field	Description
UINT8	**VideoEisSamplingRate**	The ADC sampling rate in units of milliseconds for electronic image stabilization (EIS) in video mode.
UINT8	**StillEisSamplingRate**	The ADC sampling rate in units of milliseconds for EIS in still mode.
UINT8	**MaxEisRangePerct**	Specifies maximum percentage of capture window used by EIS, ranging from 1 (1.01x d-zoom) to 99 (100X d-zoom).
UINT8	**MinEisRangePerct**	Specifies minimum percentage of capture window used by EIS, ranging from 1 (1.01x d-zoom) to 99 (100X d-zoom).
UINT8	**FlgEisDzEffect**	Enables/Disables the digital zoom effect when EIS is disabled.
UINT8	**XEisStrength**	Specifies the EIS strength in the horizontal direction, ranging from 0 (no EIS) to 99.
UINT8	**YEisStrength**	Specifies the EIS strength in the vertical direction, ranging from 0 (no EIS) to 99.
UINT8	**XRscStrength**	Specifies the rolling shutter compensation strength in the horizontal direction, ranging from 0 (disabled) to 99.
UINT8	**YRscStrength**	Specifies the rolling shutter compensation strength in the vertical direction, ranging from 0 (disabled) to 99.
UINT16	**MinSupportFr**	Specifies the minimum frames per second (FPS) supported by EIS.

*Table 2-17. Field Definition for the **AAA_PARAM_s** structure **EIS_CONTROL_s()**.*

2.3.1.1.1.17 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s

The definitions for the structure **SCENEMODE_s** are:

```
typedef struct _SCENEMODE_s_ {
UINT16  MaxTableCount;
UINT16  VideoSceneMode[8]; UINT16     StillSceneMode[8]; UINT16     VideoLv-
Lut[21]; UINT16     StillLvLut[21]; SCENE_CONTROL_s     SceneControl; SCENEMODE_
CONTROL_s     SceneModeControl;

} SCENEMODE_s;
```

The field definitions are:

Type	Parameter	Description
UINT16	**MaxTableCount**	Specifies the number of available scenes (each scene will be configured by an exclusive table).
UINT16	**VideoSceneMode**	Specifies the scene mode for the video/still mode. The software can turn off the scene mode by specifying SCENE_OFF. Available scenes are listed in `AmbaImg_AaaDef.h`.
UINT16	**StillSceneMode**	
UINT16	**VideoLvLut[21]**	Specifies the EV-index for each LV (LV 0 - 20) of the scene.
UINT16	**StillLvLut[21]**	
structure	**SceneControl**	TBD
structure	**SceneModeControl**	Please refer to **SCENEMODE_CONTROL_s** in Section 2.3.1.1.1.18 below.

*Table 2-18. Field Definition for the **AAA_PARAM_s** structure **SCENEMODE_s**.*

2.3.1.1.1.18 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s

The definitions for **SCENEMODE_CONTROL_s** are:

```
typedef struct _SCENEMODE_CONTROL_s_ {
  SC_LIGHT_CONDITION_s    LightCondition[5];
  int   DetectPriority[36];
  UINT8 SceneDetectNo;
  UINT8 SkipFrames;
  SCENE_DETECT_CONDITION_s      DetectCondition[36];
} SCENEMODE_CONTROL_s;
```

The field definitions are as follows:

Type	Parameter	Description
structure	**LightCondition[5]**	Please refer to **SC_LIGHT_CONDITION_s** in Section 2.3.1.1.1.19 below.
int	**DetectPriority[36]**	These parameters are used to list the order for the scene detection.
UINT8	**SceneDetectNo**	Specifies the numbers of the scenes for the auto scene detection.
UINT8	**SceneFrames**	This parameter is used to set the skip frames for the scene detection.
structure	**DetectCondition[36]**	Please refer to **SCENE_DETECT_CONDITION_s** in Section 2.3.1.1.1.21 below.

*Table 2-19. Field Definition for the **AAA_PARAM_s** structure **SCENE_CONTROL_s()**.*

2.3.1.1.1.19 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SC_LIGHT_CONDITION_s

The definitions for **SC_LIGHT_CONDITION_s** are:

```
typedef struct _SC_LIGHT_CONDITION_s_ {
    HISTO_INFO_s    AutoKnee;
    HISTO_INFO_s    Gamma;
    HISTO_INFO_s    LExpo;
} SC_LIGHT_CONDITION_s;
```

The field definitions are:

Type	Field	Description
structure	**AutoKnee**	Please refer to **SC_LIGHT_CONDITION_s** in Section 2.3.1.1.1.20 below.
structure	**Gamma**	
structure	**LExpo**	

*Table 2-20. Field Definition for the **AAA_PARAM_s** structure **SC_LIGHT_CONDITION_s()**.*

2.3.1.1.1.20 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SC_LIGHT_CONDITION_s > HISTO_INFO_s

The definitions for **HISTO_INFO_s** are:

```
typedef struct _HISTO_INFO_s_ {
    UINT8     Enable;
    UINT16    StartLvl;
    UINT16    EndLvl;
    UINT16    HistoMinNo;
    UINT16    HistoMidMinNo;
    UINT16    HistoMidMaxNo;
    UINT16    HistoMaxNo;
} HISTO_INFO_s;
```

The field definitions are:

Type	Field	Description
UINT8	**Enable**	ENABLE DISABLE This parameter is used to enable/disable this kind of function.
UINT16	**StartLvl**	These parameters are used to configure the effective histogram range (Figure 2-2). The scale of these parameters (the brightness) is 0 - 255. Because the total number of the full histogram will be scaled to be 4096, the output between the **start_lvl** and **end_lvl** will be 0 - 4096.
UINT16	**EndLvl**	
UINT16	**HistoMinNo (0 ~ 4096)**	These parameters are used to define the AREA-A, AREA- B, AREA-C, AREA-D, and AREA-E in Figure 2-3. The scale of these parameters is 0 - 4096. The four scene modes can dynamically control the gamma, local exposure, and auto-knee parameters. If the value is in AREA-A, the ratio will apply the mini one. If the value is in AREA-C, the ratio will apply the mid one. If the value is in AREA-E, the ratio will apply the max one. If the value is in AREA-B, the ratio will apply the interpolation of the minimum and the middle scale of parameters. If the value is in AREA-D, the ratio will apply the interpolation of the middle and the maximum scale of parameters.
UINT16	**HistoMidMinNo (0 ~ 4096)**	
UINT16	**HistoMidMaxNo (0 ~ 4096)**	
UINT16	**HistoMaxNo (0 ~ 4096)**	

*Table 2-21. Field Definition for the **AAA_PARAM_s** structure **HISTO_INFO_s()**.*

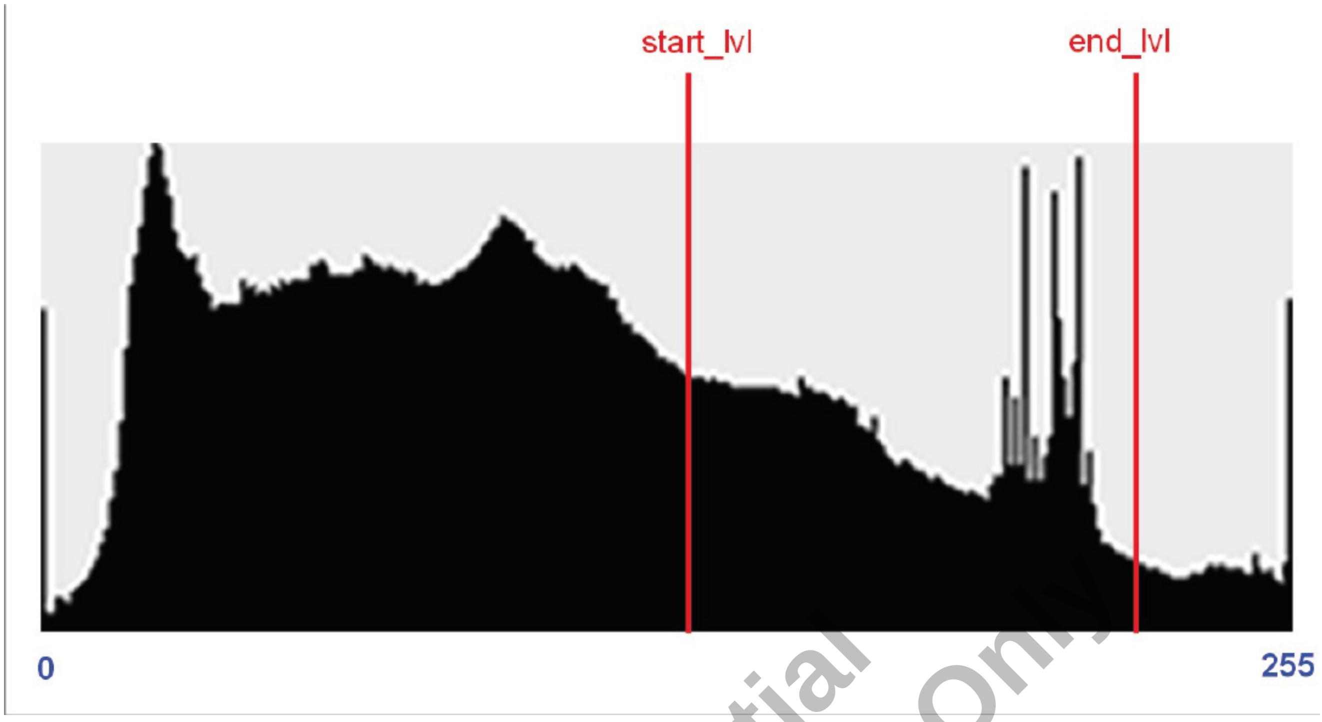


*Figure 2-2. The Effective Histogram for the Structure **HISTO_INFO_s**.*

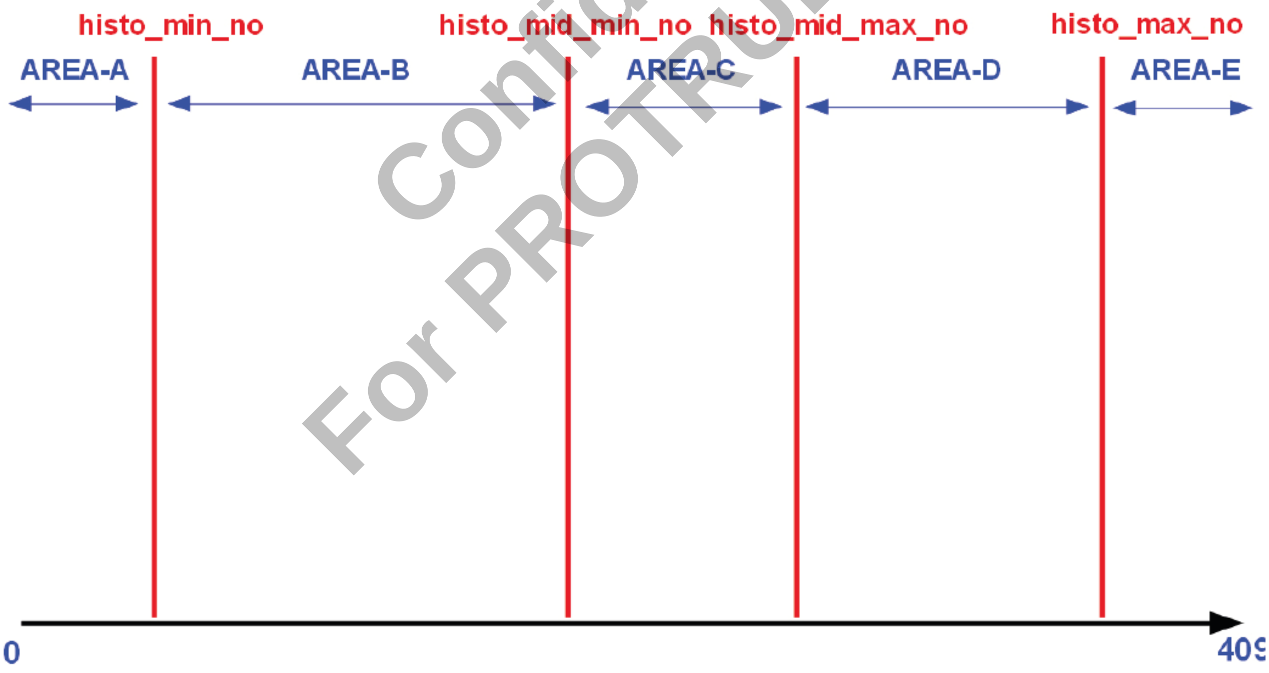


*Figure 2-3. The Areas for the Structure **HISTO_INFO_t**.*

2.3.1.1.1.21 AmbaIQParamXXX_DefaultParams.c > AAA_PARAM_s > SCENEMODE_s > SCENEMODE_CONTROL_s > SCENE_DETECT_CONDITION_s

The definitions for **SCENE_DETECT_CONDITION_s** are:

```
typedef struct _SCENE_DETECT_CONDITION_s_ {
    UINT8               Face;
    UINT16              LvNoMin;
    UINT16              LvNoMax;
    UINT16              DistanceMin;
    UINT16              DistanceMax;
    UINT16              MotionMin;
    UINT16              MotionMax;

    UINT16              LightLow;
    UINT16              LowThreshold;
    UINT16              LightHigh;
    UINT16              HighThreshold;
} SCENE_DETECT_CONDITION_s;
```

The field definitions are:

Type	Field	Description
UINT8	**Face**	0: TBD 1: Use the face detection to be the scene detection term.
UINT16	**LvNoMin**	When the ambient brightness is higher than the **LvNoMin** , this field will return success.
UINT16	**LvNoMax**	When the ambient brightness is lower than the LvNoMax , this field will return success.
UINT16	**DistanceMin**	When the focus distance (should be supported by the lens driver) is longer than the **DistanceMin**, this field will return success.
UINT16	**DistanceMax**	When the focus distance (should be supported by the lens driver) is shorter than the **DistanceMax**, this field will return success.
UINT16	**MotionMin**	When the motion value is higher than the **MotionMin**, this field will return success.
UINT16	**MotionMax**	When the motion value is smaller than the **MotionMax**, this field will return success.
UINT16	**LightLow**	If the sum of the histogram between 0 - LowThreshold (in the scale of 0 - 255) is bigger than **LightLow** (the scale of full histogram summation is 4096), this field will return success. Please refer to Figure 2-4.
UINT16	**LowThreshold**	
UINT16	**LightHigh**	If the sum of the histogram between HighThreshold (in the scale of 0 ~ 255) - 255 is bigger than **LightHigh** (the scale of full histogram summation is 4096), this field will return success. Please refer to Figure 2-5.
UINT16	**HighThreshold**	

*Table 2-22. Field Definition for the **AAA_PARAM_s** structure **SCENE_DETECT_CONDITION_s()**.*

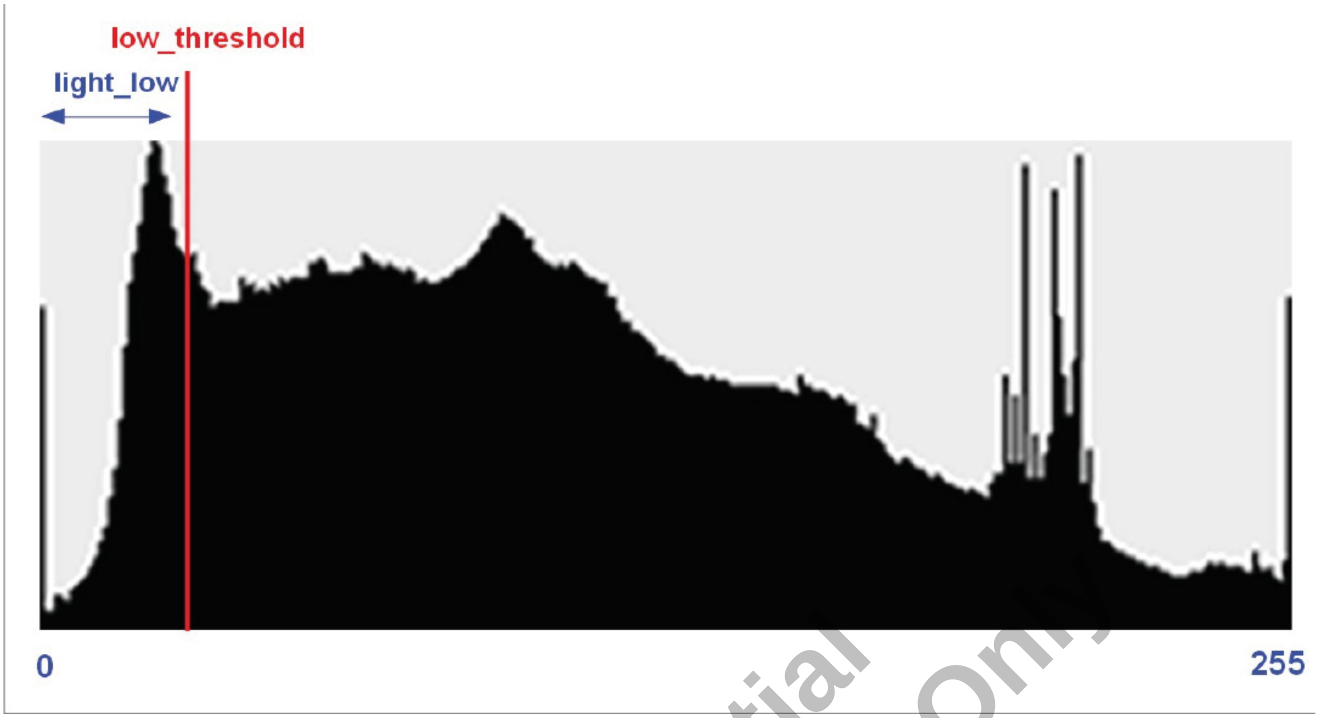


Figure 2-4. LightLow and ***LowThreshold*** *for the Structure* ***SCENE_DETECT_CONDITION_s****.*

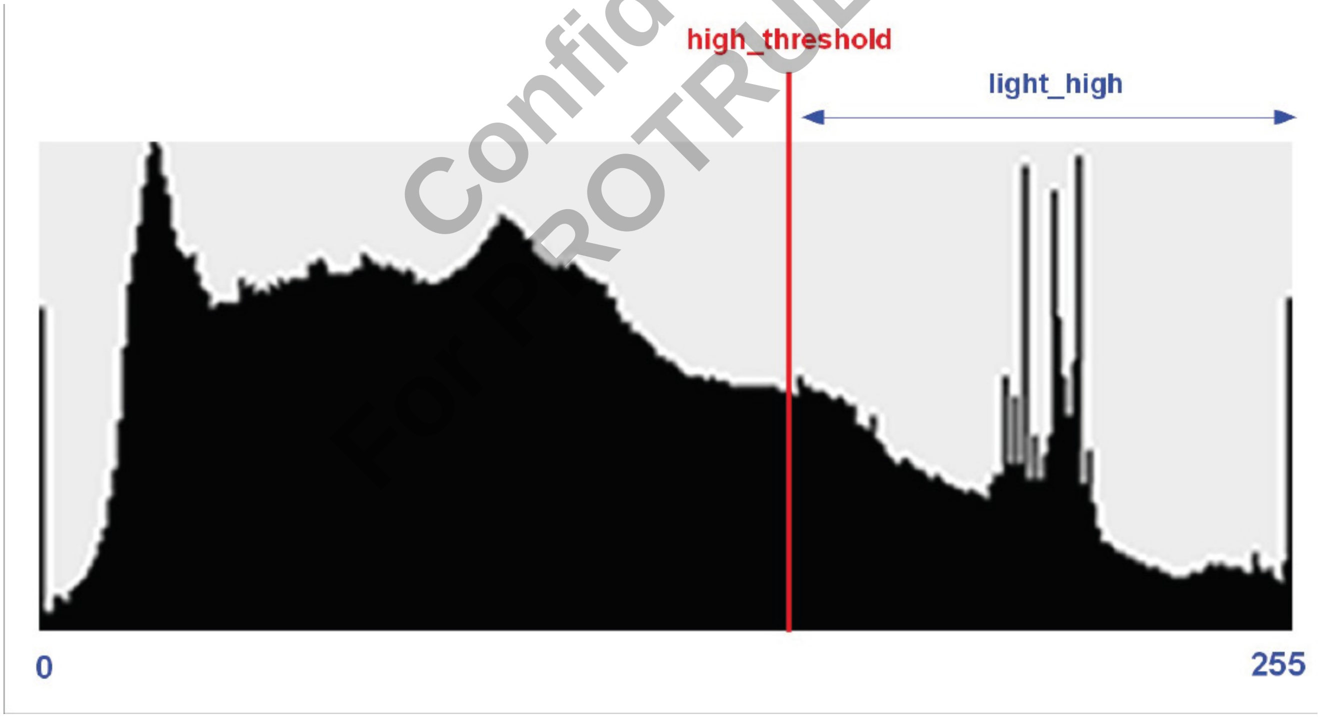


Figure 2-5. ***LightHigh*** *and* ***HighThreshold*** *for the Structure* ***SCENE_DETECT_CONDITION_s****.*

2.3.2 Header File Details: AmbaIQParamXXX_A12_ADJ_VideoXXX.c

The header file `AmbaIQParamXXX_A12_Adj_VideoXXX.c` contains **AmbaIQParamXXX_AdjVideoPc[1]**, which is used for PC-mode video IQ tuning. The file `AmbaIQParamXXX_A12_Adj_StillLIso.c` contains **AmbaIQParamXXX_AdjStillLIso[1]**, which is used for still low ISO image IQ tuning. The file `AmbaIQParamXXX_A12_Adj_StillHIso.c` contains **AmbaIQParamXXX_AdjStillHIso[1]**, which is used for still high ISO image IQ tuning.

Type	Field	Description
AmbaIQParamXXX_A12_Adj_VIDEO /PhotoPreviewXXX c	**ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s**	**AmbaIQParamXXX_AdjVideoPc[1]**/`AmbaIQParamXXXAdjPhotoPreview.c`. Please refer to Section 2.3.2.1.1.
AmbaIQParamXXX_A12_Adj_StillLIso.c	**ADJ_STILL_FAST_LISO_PARAM_S**	**AmbaIQParamXXX_AdjStillLIso[1]** Please refer to Section 2.3.3.1.1.
AmbaIQParamXXX_A12_Adj_StillHIso.c	**ADJ_STILL_FAST_HISO_PARAM_S**	**AmbaIQParamXXX_AdjStillHIso[1]** Please refer to Section 2.3.3.1.1.

*Table 2-23. Header `AmbaIQParamXXX_A12_Adj_XXX.c` Settings to Adjust Video / Still Modes**()**.*

2.3.2.1 AmbaIQParamXXX_A12_Adj_VideoPc.c : Programming Map

- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > ADJ_AWB_AE_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > ADJ_AWB_AE_s > ADJ_LUT_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s>ADJ_FILTER_INFO_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s>ADJ_DEF_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > COLOR_3D_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > COLOR_3D_s > ADJ_COLOR_CONTROL_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > ShutterBlackLevel[ADJ_NF_TABLE_COUNT]
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > BlackLevel[ADJ_NF_TABLE_COUNT]
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_BASIC_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > LowMctfA[25], HighMctfA[25], LowMctfB[25], HighMctfB[25]
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s > FIR_COEFS_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s
- AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s > AMBA_DSP_IMG_CORING_s

2.3.2.1.1 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s and ADJ_PHOTO_PARAM_s

The `AmbaIQParamXXX_A12_ADJ_VideoPc.c` data structure **ADJ_VIDEO_PARAM_s** is:

```
    typedef struct _ADJ_PHOTO_PARAM_s_ {
      UINT32               VersionNumber;
      UINT32               ParamVersionNum;
      ADJ_AWB_AE_s         NormalAwbAe;   //ADJ DEF
      ADJ_AWB_AE_s         FlashAwbAe;    //ADJ DEF
      VIDEO_FILTER_PARAM_s FilterParam;
    } ADJ_PHOTO_PARAM_s;
```

The field definitions are:

Type	Field	Description
UINT32	**VersionNumber**	Version number
UINT32	**ParamVersionNum**	Version number of these tuning parameters
structure	**NormalAwbAe**	These settings work for the photo-preview mode and still structure. Please refer to **ADJ_AWB_AE_s** in Section 2.3.2.1.1.1.
structure	**FlashAwbAe**	
structure	**FilterParam**	Please refer to **VIDEO_FILTER_PARAM_s** in Section 2.3.2.1.1.3.

*Table 2-24. Field definition for the `AmbaIQParamXXX_A12_Adj_PhotoPreview.c` structure **ADJ_PHOTO_PARAM_s()**.*

2.3.2.1.1.1 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > ADJ_AWB_AE_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **ADJ_AWB_AE_s** defines the R and B ratio of AWB gain to fine tune the color appearance after AWB. The AE target is also specified here.

```
    typedef struct _ADJ_AWB_AE_s_
      { UINT8     MaxTableCount;
      ADJ_LUT_s   Table[25];
    } ADJ_AWB_AE_s;
```

The field definitions are as follows:

Type	Field	Description
UINT8	**MaxTableCount**	Specifies the maximum number of ev-index levels. The R, B ratio and AE target for each ev-index level can be fine-tuned.
structure	**Table[25]**	Please refer to **ADJ_LUT_s** in Section 2.3.2.1.1.2 below.

*Table 2-25. Header `AmbaIQParamXXX_A12_Adj_XXX.c` Settings to Adjust Video / Still Modes**()**.*

2.3.2.1.1.2 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > ADJ_AWB_AE_s > ADJ_LUT_s

The data structure **ADJ_LUT_s** defines the 16 values available in each of the 25 tables used with **ADJ_AWB_AE_s**, for example. Currently seven values are in use, six of them are for R and B ratio of AWB gain at low color-temperature, D50, and high color-temperature condition. The AE target of a specific EV-index level is the 7th value.

The exposure value (EV) index using the official AE algorithm, is a combination of shutter time, gain, and iris. The EV index use the official AE algorithm and can be interpreted as follows:

1. Larger EV index means darker ambient brightness.
2. Smaller EV index means brighter ambient brightness.

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **ADJ_LUT_s** is:

```
typedef struct _ADJ_LUT_s_ {
  INT16 Value[24];
} ADJ_LUT_s;
```

Example **ADJ_LUT_s** settings for video adjustment are as follows:

//	AWB R & B	ratio					
// low_temp_target		D50_target		high_temp_target		// AE target	
//R	B	R	B	R	B		
{128,	128,	124,	116,	132,	108,	150},	//ev-index 0
{128,	128,	124,	116,	132,	108,	150},	//ev-index 1024
{128,	128,	124,	116,	132,	108,	150},	//ev-index 2048
...					...		
...					...		
{128,	128,	124,	116,	132,	108,	150},	//ev-index

*Table 2-26. Example **ADJ_LUT_s** settings.*

These values are used in the above structure to set the WB ratio and AE target. The AWB ratio controls the color tone. According to the AWB estimate, if the light source is located at the low-color temperature the final WB gain will be the product of the estimated gain and the **low_temp_target**. If the estimated light source is located at the high color-temperature, the final WB gain will be the product of the estimated gain and the **high_temp_target**. If the estimated light source is located between D50 and high color temperature, the final gain will be the product of the estimated gain and the interpolation of the d50_target and the **high_temp_target**. The scale of the **low_temp_target**, d50_target, and the **high_temp_target** is 128.

2.3.2.1.1.3 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **VIDEO_FILTER_PARAM_s** is:

```
typedef struct _VIDEO_FILTER_PARAM_s_ {
  ADJ_FILTER_INFO_s EvImg;      //ADJ DEF
  UINT8 NfMaxTableCount;
  ADJ_DEF_s   Def;  //ADJ DEF

  ADJ_BASIC_s Basic;
  UINT8 MctfEnable;
  ADJ_LUT_s   LowMctfA[25];
  ADJ_LUT_s   HighMctfA[25];
  ADJ_LUT_s   LowMctfB[25];
  ADJ_LUT_s   HighMctfB[25];
  ADJ_LUT_AGC_WB_s  ChromaFilter;

  DEF_SHARP_INFO_s  SharpInfo;
} VIDEO_FILTER_PARAM_s
```

The field definitions are:

Type	Field	Description
structure	**EvImg**	Please refer to **ADJ_FILTER_INFO_s** in Section 2.3.2.1.1.4 below.
UINT8	**NfMaxTableCount**	Specifies the maximum number of nf-index levels. Software can fine tune the black-level correction, bad-pixel correction, and CFA filtering for each nf-index level.
structure	**Def**	Please refer to **ADJ_DEF_s** in Section 2.3.2.1.1.5
structure	**Basic**	Please refer to **ADJ_BASIC_s** in Section 2.3.2.1.1.10
UINT8	**MctfEnable**	This value is used to enable/disable the MCTF (This a filter block in our IDSP) filter. 0: DISABLE 1: ENABLE
structure	**LowMctfA[25]**	Set MCTF parameters for low-color temperature. Please refer to description in Section 2.3.2.1.1.11.
structure	**HighMctfA[25]**	Set MCTF parameters for high-color temperature. Please refer to description in Section 2.3.2.1.1.11.
structure	**LowMctfB[25]**	Set MCTF parameters for low-color temperature. Please refer to description in Section 2.3.2.1.1.11.
structure	**HighMctfB[25]**	Set MCTF parameters for high-color temperature. Please refer to description in Section 2.3.2.1.1.11.
structure	**ChromaFilter**	These tables can be used to set the chroma filter parameters according to the color temperature.
structure	**SharpInfo**	Please refer to **DEF_SHARP_INFO_s** in Section 2.3.2.1.1.12 below.

*Table 2-27. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` structure **VIDEO_FILTER_PARAM_s()**.*

2.3.2.1.1.4 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s>ADJ_FILTER_INFO_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **ADJ_FILTER_INFO_s** is:

```
    typedef struct _ADJ_FILTER_INFO_s_ {
        UINT8        TableCount;
        ADJ_LUT_s    Enable;
        ADJ_LUT_s    EvTable[25];
        ADJ_LUT_s    NfTable[25];
    } ADJ_FILTER_INFO_s;
```

The definitions for the fields above are as follows:

Type	Field	Description
UINT8	**TableCount**	Specifies the maximum number of ev-index levels. Software can fine-tune the luminance offset, color saturation, gamma, local-exposure, and turn on anti-aliasing for each ***ev-index*** level.
structure	**Enable**	The enable field holds flags enabling the tuning parameters defined in **ev_img**. An example of flag settings is provided below. Currently, 8 out of 16 values are used.
structure	**EvTable[25]**	There are 16 values available in each of the 25 tables. Currently, 11 values are in use: Y-offset, UV saturation ratio, color ratio, gamma ratio, local-exposure ratio, chroma-scale ratio, anti-aliasing strength, and **gbgr_mismatch narrow_enable**, wide_enable, safety, and thresh. These can be tuned based on the **ev_index**.
structure	**NfTable[25]**	There are 16 values available in each of the 25 tables. Currently, 7 values are in use: Y-offset, UV saturation ratio, color ratio, gamma ratio, local-exposure ratio, chroma-scale ratio, and anti-aliasing strength. These can be tuned based on the **nf_index**.

*Table 2-28. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` Structure **ADJ_FILTER_INFO_s()**.*

Example: **ADJ_LUT_s** settings to enable **ev_img** tuning parameter flags are:

//	enable of		ev_img					
// Y-offset	UV	Color	Gamma	local-ex- posure	chroma-scale	anti- alias		
{0,	64,	128	128	128	60	0}	//ev- index	0
{0,	64,	128	128	128	60	1}	//ev- index	1024
{0,	64,	128	128	128	60	2}	//ev- index	2048
...								
...								
{0,	64,	128	128	128	60	3}	//	

*Table 2-29. Example **ADJ_LUT_s** Settings.*

Each value takes effect only if the corresponding flag in enable field is set as 1.

The definitions for the fields above are as follows:

Field	Description
Y-offset	Sets the Y-offset in the RGB_to_YUV matrix.
UV saturation ratio	Modifies the UV settings in the **RGB_to_YUV** matrix. The scale is 64. If the value is higher than 64, the color will be more vivid than the standard **RGB_to_ YUV** matrix. If the value is less than 64, the color will be less saturated than the standard **RGB_to_YUV** matrix.
color ratio	Generates the mixed color correction function in the formula shown below: When color_ratio is between 0~128 Final_CC = (Internal_CC * color_ratio + CC3 * (128 - color_ratio)) / 128 When color_ratio is between 128 ~ 256 Final_CC = (Internal_CC*(256-color_ratio) + CC4*(color_ratio-128)) / 128 Internal_CC: The color correct is decided by the color temperature CC3, CC4: Please refer to Section 2.4.
gamma ratio	Controls the gamma-curve and the valid range is from 0 to 255. The final gamma- curve is determined by the ratio and the two curves specified by **ratio_255_gamma** and **ratio_0_gamma**. A value of 128 implies no effect.
local-exposure ratio	Sets the local exposure curve, the valid range is from 0 to 255. The final local-exposure curve is determined by the ratio and two curves specified by **l_expo_255** and **l_expo_0**. A value of 128 implies no effect.
chroma-scale ratio	Sets the chroma scale curve, the valid range is from 0 to 255. The final local-exposure curve is determined by the ratio and two curves specified by **chroma_curve_255** and **chroma_curve_0**. A value of 128 implies no effect.
anti-aliasing strength	Specifies the strength of the anti-aliasing used to remove the artifacts occurring around the edges and high-frequency areas due to sharpening. The range of the value is from 0 to 3 (3 = maximum strength).
gbgr_mismatch narrow_enable	0: Disable 1: Enable, Detects if there is very consistent mismatch in a small area (if narrow_enable = 1).
gbgr_mismatch wide_enable	0: Disable 1: Enable, Detects if there is mismatch in a wide area (if wide_enable = 1).

Field	Description
gbgr_mismatch wide_thresh	0-256, Wide detection passes if both of following statements are true: - A measure of mismatch is greater than **wide_thresh**, so increasing wide_thresh make the filter weaker. - A measure of how the likely system mismatch is caused by the true signal is less than **wide_safety**.
gbgr_mismatch wide_safty	0-256, increasing **wide_safety** makes the filter stronger.

*Table 2-30. Detail from `AmbaIQParamXXX_A12_Adj_XXX.c`: The **ADJ_LUT_s** > enable flags for **ADJ_FILTER_INFO_s** > **ev_img**.*

2.3.2.1.1.5 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s>ADJ_DEF_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **ADJ_DEF_s** is:

```
    typedef struct _ADJ_DEF_s_ {

        COLOR_3D_s                   Color;
        UINT8                        BlackLevelEnable;
        UINT16                       StartShutterIdx;
        UINT16                       ShutterTableNo;
        ADJ_LUT_s                    ShutterBlackLevel[25];
        ADJ_LUT_s                    BlackLevel[25];
        AMBA_DSP_IMG_TONE_CURVE_s    Ratio255Gamma;
        AMBA_DSP_IMG_TONE_CURVE_s    Ratio0Gamma;
        UINT16                       LExpo255[NUM_EXPOSURE_CURVE];
        UINT16                       LExpo0[NUM_EXPOSURE_CURVE];
        UINT16                       ChromaCurve255[NUM_CHROMA_GAIN_CURVE];
        UINT16                       ChromaCurve0[NUM_CHROMA_GAIN_CURVE];
    } ADJ_DEF_s;
```

The field definitions are:

Type	Field	Description
structure	**Color**	Please refer to **COLOR_3D_s** in Section 2.3.2.1.1.6.
UINT8	**BlackLevelEnable**	Turn on/off the black-level correction: 0: DISABLE 1: ENABLE
UINT16	**StartShutterIdx**	Specifies the starting shutter index for black level correction, that is the longest shutter time.
UINT16	**ShutterTableNo**	Specifies the number of black level correction tables. The first black level correction tables is for the longest shutter time as described above.
structure	**ShutterBlackLevel [25]**	This field defines the black level corrections for each shutter time (a maximum of 25 levels available). The first level and the number of levels are defined by **start_shutter_idx** and **shutter_table_no**. The definitions for **ShutterBlackLevel[25]** are provided in Section 2.3.2.1.1.8. Also, please refer to **ADJ_LUT_s**.

Type	Field	Description
structure	**BlackLevel[25]**	This parameter defines the offset values for black level correction (BLC) according to Noise Filter Index (nf-index) level, color temperature, and color channel. The definitions for **BlackLevel[25]** are provided below in Section 2.3.2.1.1.9. Also, please refer to **ADJ_LUT_s** .
structure	**Ratio255Gamma**	Please refer to **gamma_curve_info_t** in the structure definition. Specifies the gamma tables for the R, G, and B channels corresponding to a gamma-ratio of 255. The dimensions of each table are 1x256.
structure	**Ratio0Gamma**	Please refer to **gamma_curve_info_t** in the structure definition. Specifies the gamma tables for R, G, and B channels corresponding to a gamma-ratio of 0. The dimensions of each table are 1x256. The effects of **ratio_255_gamma** and **ratio_0_ gamma** will be dominated by the gamma-ratio, please refer to **ev_img** for details.
UINT16	**LExpo255[NUM_EXPOSURE_CURVE]**	Specifies the local exposure curve corresponding to a local exposure-ratio of 255. The dimensions are defined by **NUM_EXPOSURE_CURVE**.
UINT16	**LExpo0[NUM_EXPOSURE_CURVE]**	Specifies the local exposure curve corresponding to a local exposure-ratio of 0. The dimensions are defined by **NUM_EXPOSURE_CURVE**.
UINT16	**ChromaCurve255[NUM_CHROMA_GAIN_CURVE]**	Specifies the chroma gain curve corresponding to a chroma-ratio of 255. The dimensions are defined by **NUM_CHROMA_GAIN_CURVE**.
UINT16	**ChromaCurve0[NUM_CHROMA_GAIN_CURVE]**	Specifies the chroma gain curve corresponding to a chroma-ratio of 0. The dimensions are defined by **NUM_CHROMA_GAIN_CURVE**.

Table 2-31.   Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` *Structure* ***ADJ_DEF_s()****.*

2.3.2.1.1.6 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > COLOR_3D_s

The AmbaIQParamXXX_A12_ADJ_XXX.c data structure **COLOR_3D_s** is:

```
typedef struct _COLOR_3D_s_ {
    UINT8                Type;
    UINT8                Control;
    ADJ_COLOR_CONTROL_s Table[5];
} COLOR_3D_s
```

The field definitions are as follows:

Type	Field	Description
UINT8	**Type**	IMG_MODE_VIDEO: Apply Video 3D-CC. IMG_MODE_STILL: Apply Still 3D-CC.
UINT8	**Control**	1: 3D Color Correction Mode
structure	**Table [5]**	Sets the color correction values. The values of the **r_gain** and **b_gain** should be modified carefully. The color temperature related parameters will refer to these values (r_gain and b_gain). Please refer to **ADJ_COLOR_CONTROL_s** in Section 2.3.2.1.1.7.

*Table 2-32. Field Definition for the AmbaIQParamXXX_A12_Adj_XXX.c Structure **ADJ_DEF_s > COLOR_3D_s()**.*

2.3.2.1.1.7 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > COLOR_3D_s > ADJ_COLOR_CONTROL_s

Structure table[5] sets the color correction values. The values of the **r_gain** and **b_gain** should be modified carefully. The black-level adjustment implements these corrections.

The **COLOR_3D_s** data structure **ADJ_COLOR_CONTROL_s** is:

```
typedef struct _ADJ_COLOR_CONTROL_s_ {
    UINT16    GainR;
    UINT16    GainB;
    UINT32    MatrixThreeDTableAddr;
    INT16     CcMatrix[9];
} ADJ_COLOR_CONTROL_s;
```

The field definitions are as follows:

Type	Field	Description
UINT16	**GainR**	The R and B gain is used to reach the correct white balance. In this case, they serve as an index to find the color temperature.
UINT16	**GainB**	
UINT32	**MatrixThreeDTableAddr**	The address of the 3D color conversion, correction, and gamma tables.
INT16	**CcMatrix [9]**	The coefficients (3x3=9) of the color correction matrix (specified but currently unused).

Table 2-33. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` *Structure **ADJ_DEF_s > COLOR_3D_s > ADJ_COLOR_CONTROL_s()**.*

2.3.2.1.1.8 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > ShutterBlackLevel[ADJ_NF_TABLE_COUNT]

The **ADJ_DEF_s** field **ShutterBlackLevel[25]** defines black level corrections for each shutter time (a maximum of 25 levels available). The first level and the number of levels are defined by **start_shutter_idx** and **shutter_table_no**. The back light compensation (BLC) values for R, Gr, Gb and B channels are given below:

```
//black-correction
//R-ch          Gr-ch       Gb-ch       B-ch
{      -290,    -295,       -275,       -290 },        // shutter_index1
{      -285,    -290,       -300,       -285 },        // shutter_index2
{      -271,    -300,       -320,       -271 },        // shutter_index3
...
{      -210,    -300,       -280,       -210 }},       // shutter_index10
```

Table 2-34. BLC Values for R, Gr, Gb and B Channels.

2.3.2.1.1.9 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_DEF_s > BlackLevel[ADJ_NF_TABLE_COUNT]

The **ADJ_DEF_s** field **BlackLevel[25]** defines the offset values for BLC according to nf-index level, color temperature, and color channel. A maximum of 25 nf-index levels are available:

//	low temperature				D50				high temperature						
//	R-ch	Gr-ch	Gb-ch	B-ch	R-ch	Gr-ch	Gb-ch	B-ch	R-ch	Gr-ch	Gb-ch	B-ch			
	0,	0,	0,	0,	-10	0,	0,	-25	-10	0,	0,	-25	},//	nf-index	0
{	-20	-20	-20	-130	-30	-10	-10	-80	-30	-10	-10	-80	},//	nf-index	1024
{	-20	-20	-20	-130	-30	-10	-10	-80	-30	-10	-10	-80	}, //	nf-index	2048
{	-20	-20	-20	-130	-30	-10	-10	-80	-30	-10	-10	-80	}}, //	nf-index	9216 9216

*Table 2-35.   Example **ADJ_DEF_s** settings.*

The nf-index using the official AE algorithm is relative to gain. Larger nf index means larger gain. Smaller nf index means smaller gain.

2.3.2.1.1.10 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > ADJ_BASIC_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **ADJ_BASIC_s** is:

```
    UINT8             BadPixelEnable;
        ADJ_LUT_s         BadPixel[ADJ_NF_TABLE_COUNT];
        ADJ_LUT_s         AliasingGbGrEnable;
        ADJ_LUT_s         AliasingGbGr[ADJ_NF_TABLE_COUNT];
        UINT8             ChromaMedianEnable;
        ADJ_LUT_s         LowChromaMedian[ADJ_NF_TABLE_COUNT];
        ADJ_LUT_s         HighChromaMedian[ADJ_NF_TABLE_COUNT];
        UINT8             DemoasicEnable;
        ADJ_LUT_s         Demoasic[ADJ_NF_TABLE_COUNT];
        ADJ_LUT_AGC_WB_s CfaFilter;
    }ADJ_BASIC_s;
```

The field definitions are as follows:

Type	Field	Description
UINT8	**BadPixelEnable**	ENABLE DISABLE
structure	**BadPixel[ADJ_NF_TABLE_COUNT]**	Sets the Dynamic Bad Pixel Parameters.

Type	Field	Description
structure	**AliasingGbGrEnable**	0: Disable 1: Enable
structure	**AliasingGbGr[ADJ_NF_TABLE_COUNT]**	Please refer to **gbgr_mismatch** in Section 2.3.2.1.1.4.
UINT8	**ChromaMedianEnable**	ENABLE DISABLE
structure	**LowChromaMedian[ADJ_NF_TABLE_COUNT]**	Sets the chroma median parameters for low-color temperature. Please refer to the description below.
structure	**HighChromaMedian[ADJ_NF_TABLE_COUNT]**	Sets the chroma median parameters for high-color temperature. Please refer to the description below.
UINT8	**DemoasicEnable**	ENABLE DISABLE
structure	**Demoasic[ADJ_NF_TABLE_COUNT]**	Sets the demoasic parameters. Please refer to the description below.
structure	**CfaFilter**	Sets the color filter array (CFA) Noise Filter Parameters. The CFA filter tables are divided into two groups; one is for low-color temperatures and the other is for high-temperatures.

*Table 2-36. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` Structure **ADJ_BASIC_s()**.*

Low/High ChromaMedian define demoasic parameters as the following:

Field	Description
CbAdaptiveAmount	0 – 256.
CrAdaptiveAmount	0 – 256.
CbAdaptiveStrength	0 – 256. Chroma median filter adaptive strength for Cb/Cr component.
CrAdaptiveStrength	
CbNonAdaptiveStrength	0 – 256. Chroma median filter non-adaptive strength for Cb/ Cr component.
CrNonAdaptiveStrength	

Table 2-37. Demoasic Parameters.

The structure Demoasic defines demoasic parameters:

Field	Description
GradNoiseThresh	0-32767, Gradient noise threshold
GradClipThresh	0-4095, Gradient clip threshold
ActivityThresh	0-31, Activity threshold
ActivityDifferenceThresh	0-16383, Activity difference threshold
ZipperNoiseDifferenceAd- dThresh	0-32767, Zipper Noise Difference Add Threshold
ZipperNoiseDifferenceMult- Thresh	0-255, Zipper Noise Difference Multiply Threshold
BlackWhiteResolutionDetail	0-255, Black White Resolution Detail
ClampDirectionalCandidates	0-1, Clamp Directional Candidates

Table 2-38. Demoasic Parameters.

The CfaFilter sets the CFA noise filter parameters.

Field	Description
NoiseLevel_r, g, b	The range is 0-8192.
riginalBlendStr_r, g, b	The range is 0-256.
ExtentRegular_r, g, b	The range is 0-256.
ExtentFine_r, g, b	The range is 0-256.
StrengthFine_r, g, b	The range is 0-256.
SelectivityRegular	The range is 0-256.
SelectivityFine	The range is 0-256.

Table 2-39. CFA Noise Filter Parameters.

2.3.2.1.1.11 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > LowMctfA[25], HighMctfA[25], LowMctfB[25], HighMctfB[25]

Example for **LowMctfA[25]**:

```
  //Low-color temperature
  // Y
 // TA0      TA1      TA2        TA3      TT0    TT1    TT2    TT3          …………..
 {{  20,     64,      255,       255,     10,    20,    20,    20,    20,   …………..},   // [0]
  {  20,     64,      255,       255,     10,    20,    20,    20,    20,   …………..},   // [1]
 ……………………………….
```

The relation of the above parameters is shown in the following picture. α0, α1, α2, α3 is TA0,TA1,TA2,TA3.

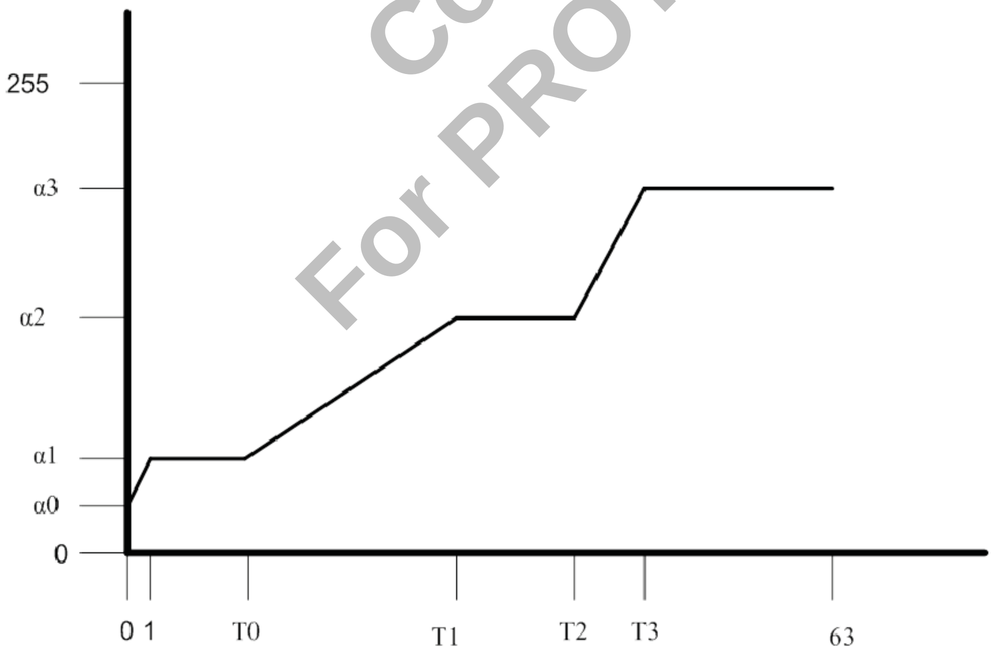


*Figure 2-6. **Low MctfB[25]**.*

Example for Low MctfB[25]:

//Low color temperature

// Y				Cb				Cr					Y	Cb		
// radius	3d	spat	adj	radius	3d	spat	adj	radius	3d	spat	adj	CmbStr	TmpMaxChg	TmpMaxChg	TmpMaxChg	
{{ 27,	85,	27,	118,	27,	85,	27,	118,	27,	85,	27,	118,	1,	100,	100,	100 },	// [0]
{ 27,	85,	27,	118,	27,	85,	27,	118,	27,	85,	27,	118,	1,	100,	100,	100 },	// [1]

*Table 2-40. Example of **Low MctfB[25]**.*

A weight (W) is computed using the following parameters,
TA0 – TA3, TT0 – TT3,

A preliminary filtered sample is computed as:

preliminary = ((256-W) * previous + W * current) / 256

The final filtered sample is computed as the preliminary filtered sample. Change from the current sample is limited to **maxchange**.

After MCTF is finished, the pre- and post-MCTF luma data is combined to make the picture look more natural. The strength of the combination is controlled by **combined_str_y**; where 0 is no effect, and larger values can make the picture look more natural and also improve the noise reduction.

strength_3D and **strength_spatial** are used to adaptively blend in the “spatial filter”.
The size (spatial extent) of the spatial filter depends on:

- Radius: (values of 0 256 correspond to 0x0 5x5)
- **level_adjust**. Larger values increase the effective radius in the dark areas as the dark areas tend to be noisier.

2.3.2.1.1.12 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s

The `AmbaIQParamXXX_A12_ADJ_XXX.c` data structure **DEF_SHARP_INFO_s** is:

```
typedef struct _DEF_SHARP_INFO_s_ {
    UINT8       CdnrEnable;
    ADJ_LUT_s   CdnrLut[25];
    ADJ_LUT_s   ShpANotAsf[25];
    UINT8       ShpAEnable;
    ADJ_LUT_s   SpatialFilter[25];
    ADJ_LUT_s   SpatialT0T1Div[25];
    ADJ_LUT_s   SpatialLevelStrAdjust[25];
    DEF_FIR_s   DefSharpAFir;
    DEF_SHARP_s DefSharpA;
    DEF_SHARP_s DefSharpB;
} DEF_SHARP_INFO_s;
```

The definitions for the fields above are as follows:

Type	Field	Description
UINT8	**CdnrEnable**	Enable/Disable CDNR.
structure	**CdnrLut[25]**	Sets **color_dependent_noise_reduction** parameter: CdnrMode: 0: Off 1: On CdnrStrength: 0 - 256
structure	**ShpANotAsf[25]**	Selects ahpA or ASF: 0: Off, 1: Asf, 2: ShpA.
UINT8	**ShpAEnable**	Enables/Disables Adj calculation of **shpA_asf**.
structure	**SpatialFilter[25]**	Sets advanced Spatial filter parameters, please refer to the description in below.
structure	**SpatialT0T1Div[25]**	Sets advanced Spatial filter T0T1_div filter parameters, please refer to the description below.
structure	**SpatialLevelStrAdjust[25]**	Sets advanced spatial_filter level_str_adjust parameters, please refer to the description below.
structure	**DefSharpAFir**	Please refer to **DEF_FIR_s** in Section 2.3.2.1.1.13.
structure	**DefSharpA**	Please refer to **DEF_SHARP_s** in Section 2.3.2.1.1.15.
structure	**DefSharpB**	Please refer to **DEF_SHARP_s** in Section 2.3.2.1.1.15.

*Table 2-41. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` Structure **DEF_SHARP_INFO_s()**.*

Example of SpatialFilter:

```
 // Advanced spatial_filter
 //     t0              t1           alpha                               dir_decide
  //   down        up   down      up   max_down   max_up    min_up      t0      t1
 {{     0,         0,    0,       0,   0,         0,        0,          0,      0},      // [0]: 0
 {      0,         0,    0,       0,   0,         0,        0,          0,      0},      // [1]: 1024
 ………………………..
```

*Table 2-42. **Low MctfB[25]**.*

Parameter	Description
T0Down	0 - 252
T0Up	0 - 252
T1Down	1 - 254, T1 must be >= T0
T1Up	1 - 254, T1 must be >= T0
AlphaMaxDown	0 - 8
AlphaMaxUp	0 - 8
AlphaMinDown	0 - 8
AlphaMinUp	0 - 8
dir_decide t0	0 - 255
dir_decide t1	0 - 255

*Table 2-43. Example of **SpatialFilter**.*

Example of **SpatialT0T1Div**:

```
// advanced spatial_filter  t0t1_div
// low              low_   low_str   mid_   meth-   high   high_dlt    high_str
                    dlt              str    od
{{     0,           0,     0,        0,     0,      0,     0,          0,             // [0]: 0
{      0,           0,     0,        0,     0,      0,     0,          0,             // [1]: 1024
…………………………..
```

Table 2-44. ***SpatialT0T1Div.***

Parameters of the advanced spatial filter **T0T1_div**:

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 255
MidStrength	0 - 255
High	0 - 255
HighDelta	0 - 7
HighStrength	0 - 255
Method	0 -1

Table 2-45. ***T0T1_div.***

Example of **SpatialLevelStrAdjust**:

```
// advanced spatial_filter level_str_adjust                               max_change
// low          low_   low_str   mid_   meth-   high   high_dlt  high_str      down     up
                dlt              str    od
{{     0,       0,      0,       0,     0,      0,     0,        0,            0,       0 },   // [0]: 0
{      0,       0,      0,       0,     0,      0,     0,        0,            0,       0 },   // [1]: 1024
{      0,       0,     0,        0,     0,      0,     0,        0,            0,       0 },    // [2]: 2048
```

Table 2-46. ***SpatialLevelStrAdjust.***

Parameters of advanced spatial filter **level_str_adjust**:

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 64
MidStrength	0 - 64
Method	0-1
High	0 - 255
HighDelta	0 - 7
HighStrength	0- 64
Max_change down	0- 255
Max_change up	0- 255

Table 2-47. ***level_str_adjust***.

2.3.2.1.1.13 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s

```
typedef struct _DEF_FIR_s_ {
    ADJ_LUT_s    FirDirAmtStr[25];
    ADJ_LUT_s    FirIsoStr[25];
    FIR_COEFS_s FirCoefs[25];
} DEF_FIR_s;
```

Type	Field	Description
structure	**FirDirAmtStr[25]**	Sets **fir_per_dir_fir_dir_amounts** parameters, 0-256, (specify_firs=3); Sets **fir_per_dir_fir_dir_strengths** parameters, 0 - 256. (specify_firs=3).
structure	**FirIsoStr[25]**	Sets Finite impulse response (FIR), **fir_per_dir_fir_iso_strengths** parameters.
structure	**FirCoefs[25]**	Please refer to **FIR_COEFS_s** in Section 2.3.2.1.1.14.

*Table 2-48. Field Definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` Structure **VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s()**.*

Example of **FirIsoStr**:

//advanced spatial_filter, fir

// specify	strength_dir	strength_iso	wide_edge_detect	fir_per_dir_fir_iso_strengths									
{{ 2,	128,	30,	2,	0,	0,	0,	0,	0,	0,	0,	0,	0}	// [0]: 0
{ 2,	128,	30,	2,	0,	0,	0,	0,	0,	0,	0,	0,	0}	// [1]:100024
{ 2,	128,	30,	2,	0,	0,	0,	0,,	0,	0,	0,	0,	0}	// [2]:2048

Table 2-49. ***FirIsoStr***.

Parameters of advanced spatial filter FIR:

Parameter	Description
Fir specify	0 - 4
Strength_dir	0 - 256 (specify_firs=2).
Strength_iso	0 - 256 (specify_firs=0,2).
Wide_edge_detect	0 – 8: Determines how wide of an area to use when determining an edge direction; the higher the value, the wider the area used.
fir_per_dir_fir_iso_strengths	0 - 256: Set **fir_per_dir_fir_iso_strengths** parameters (specify_firs=3).

Table 2-50. Advanced Spatial Filter fir.

Description of **fir_specify** is as follows:

fir_specify	Directions	params used	Description
0	ISO only	**fir_strength_iso**	Single strength determines FIR size.
1	ISO only	**fir_coefs**	Only isotropic but fully manual.
2	ISO + dir	**fir_strength_iso fir_strength_dir**	One strength for isotropic, one for directional.
3	ISO + dir	**fir_per_dir_fir_iso_strengths fir_per_dir_fir_dir_strengths fir_per_dir_fir_dir_amounts**	For each direction, the user specifies an isotropic strength, a directional strength, and amount needed to blend isotropic and directional strengths.
4	ISO + dir	**fir_coefs**	Fully manual

*Table 2-51. **fir_specify**.*

2.3.2.1.1.14 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s > FIR_COEFS_s

```
typedef struct _FIR_COEFS_s_ {
    UINT8 Coefs[9*25];
}FIR_COEFS_s;
```

Type	Field	Description
UINT8	**Coefs[9*25]**	0 - 1023, Set **fir_coefs** parameters, (specify_firs=1, 4).

Table 2-52. Field Definition for the AmbaIQParamXXX_A12_Adj_XXX.c *Structure* ***VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_FIR_s > FIR_COEFS_s()****.*

0	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	23	22	21
20	19	18	17	16	15	14
13	12	11	10	9	8	7
6	5	4	3	2	1	0

Table 2-53. ***Fir_Coefs****.*

2.3.2.1.1.15 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s

```
typedef struct _DEF_SHARP_s_ {
    ADJ_LUT_s                 FinShpNfBoth[25];
    ADJ_LUT_s                 FinShpNfNLevelStrAdjust[25];
    DEF_FIR_s                 FinShpNfN;

    DEF_FIR_s                 FinShpNfS;
    AMBA_DSP_IMG_CORING_s FinShpNfSCoring[25];
    ADJ_LUT_s                 FinShpNfSCoringIdxScale[25];
    ADJ_LUT_s                 FinShpNfSMinCoringIdxResult[25];
    ADJ_LUT_s                 FinShpNfSScaleCoring[25];

} DEF_SHARP_s;
```

Type	Field	Description
structure	**FinShpNfBoth[25]**	Sets **sharpen_noise_filter_both** parameters.
structure	**FinShpNfNLevelStrAdjust[25]**	Sets **sharpen_noise_filter_noise** parameters.
structure	**FinShpNfN**	Sets **sharpenA_noise_filter_noise** FIR parameters Please refer to **DEF_FIR_s** in Section 2.3.2.1.1.13 and Section 2.3.2.1.1.14.
structure	**FinShpNfS**	Sets **sharpenA_noise_filter_sharpen** FIR parameters Please refer to **DEF_FIR_s** in Section 2.3.2.1.1.13 and Section 2.3.2.1.1.14.
structure	**FinShpNfSCoring[25]**	Please refer to **AMBA_DSP_IMG_CORING_s** in Section 2.3.2.1.1.16.
structure	**FinShpNfSCoringIdxScale[25]**	Sets sharpenA_noise_filter_sharpen coring_index_scale parameters. Scales the index way from the center. Strengths <16 moves the index toward the center (entry 128) and strength > 16 moves the index away from the center.

Type	Field	Description
structure	**FinShpNfSMinCoringIdxResult[25]**	Sets sharpenA_noise_filter_sharpen min_coring_result parameters. The minimum coring multiplier is the result of sharpening_min_coring_result.*/8.
structure	**FinShpNfSScaleCoring[25]**	Sets sharpenA_noise_filter_sharpen scale_coring parameters. The coring multiplier is multiplied based on the result of the sharpening_scale_coring.*.

Table 2-54. Field Definition for the `AmpIQParamXXX_A12_Adj_XXX.c` *Structure* ***VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s()****.*

Example of **sharpen_noise_filter_both** parameters:

```
//sharpenA_noise_filter_both
//                        wide               max_change_5x5   max_change
//edge_th_enable          edge_detect        up       down    up       down        mode
{{  200,       0,         0,                 0,       0,      0,       0,          2 },     // [0]: 0
{   200,       0,         0,                 0,       0,      0,       0,          2 },     // [1]: 1024
{   200,       0,         0,                 0,       0,      0,       0,          2 },     // [2]: 2048
```

Table 2-55. ***sharpen_noise_filter_both****.*

Parameter	Description
EdgeThresh	0 - 2047
Enable	0 - 1
WideEdgeDetect	0 - 8
MaxChangeUp5x5	0 - 255
MaxChangeDown 5x5	0 - 255
MaxChangeUp	0 - 255
MaxChangeDown	0 - 255
mode	0 - 2

Table 2-56. ***sharpen_noise_filter_both****.*

Example of **sharpen_noise_filter_noise** parameters:

```
//sharpenA_noise_filter_noise
//         level_str_adjust                                              max_change
//low       low_dlt    low_str    mid_    method    high     high_str    Down     Up
                                  str
{{ 0,       0,         0,         0,      0,        0,       0,          0,       0 },       // [0]: 0
{  0,       0,         0,         0,      0,        0,       0,          0,       0 },       // [1]: 1024
{  0,       0,         0,         0,      0,        0,       0,          0,       0 },       // [2]: 2048
```

Table 2-57. ***sharpen_noise_filter_both****.*

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 416
MidStrength	0 - 416
Method	0-1
High	0 - 255
HighDelta	0 - 7
HighStrength	0- 416
Max_change down	0- 255
Max_change up	0- 255

Table 2-58. ***sharpen_noise_filter_noise****.*

Example of **FinShpNfSCoringIdxScale**:

```
//sharpenA_noise_filter_noise
//            coring_index_scale
//low        low_dlt   low_str   mid_str   method      high       high_dlt  high_str
{{  0,        0,        0,        0,        0,          0,         0,        0  },         // [0]: 0
{   0,        0,        0,        0,        0,          0,         0,        0  },         // [1]: 1024
{   0,        0,        0,        0,        0,          0,         0,        0  },         // [2]: 2048
……………
```

Table 2-59. ***FinShpNfSCoringIdxScale.***

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 255
MidStrength	0 - 255
High	0 - 255
HighDelta	0 - 7
HighStrength	0- 255
Method	0- 1

Table 2-60. ***FinShpNfSCoringIdxScale.***

Example of **FinShpNfSMinCoringIdxResult**:

```
//sharpenA_noise_filter_noise
//          min_coring_index_scale
//low        low_dlt   low_     mid_str  meth-   high       high_dlt  high_str
                       str               od
{{  0,        0,        0,       0,       0,      0,         0,        0  },        // [0]: 0
{   0,        0,        0,       0,       0,      0,         0,        0  },        // [1]: 1024
{   0,        0,        0,       0,       0,      0,         0,        0  },        // [2]: 2048
……………
```

Table 2-61. ***FinShpNfSMinCoringIdxResult.***

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 255
MidStrength	0 - 255
High	0 - 255
HighDelta	0 - 7
HighStrength	0- 255
Method	0- 1

Table 2-62. ***FinShpNfSMinCoringIdxResult****.*

Example of **FinShpNfSScaleCoring**:

```
//sharpenA_noise_filter_noise
//          scale_coring
//low       low_dlt   low_str   mid_str   method   high     high_dlt   high_str
{{  0,      0,        0,        0,        0,       0,       0,         0  },        // [0]: 0
{   0,      0,        0,        0,        0,       0,       0,         0  },        // [1]: 1024
{   0,      0,        0,        0,        0,       0,       0,         0  },        // [2]: 2048
…………….
```

Table 2-63. ***Example of FinShpNfSScaleCoring****.*

Parameter	Description
Low	0 - 255
LowDelta	0 - 7
LowStrength	0 - 255
MidStrength	0 - 255
High	0 - 255
HighDelta	0 - 7
HighStrength	0- 255
Method	0- 1

Table 2-64. ***Example of FinShpNfSScaleCoring****.*

2.3.2.1.1.16 AmbaIQParamXXX_A12_Adj_XXX.c > ADJ_VIDEO_PARAM_s/ADJ_PHOTO_PARAM_s > VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s > AMBA_DSP_IMG_CORING_s

```
typedef struct _AMBA_DSP_IMG_CORING_s_ {
    UINT8   Coring[AMBA_DSP_IMG_NUM_CORING_TABLE_INDEX];
} AMBA_DSP_IMG_CORING_s;
```

Type	Field	Description
UINT8	**Coring[AMBA_DSP_IMG_NUM_CORING_TABLE_INDEX]**	0-31. Defines the coring table for the sharpening filter for each level. The dimensions of each of the 25 coring tables are 1x256. Higher coring values imply higher sharpness.

*Table 2-65. Field definition for the `AmbaIQParamXXX_A12_Adj_XXX.c` structure **VIDEO_FILTER_PARAM_s > DEF_SHARP_INFO_s > DEF_SHARP_s > AMBA_DSP_IMG_CORING_s ()**.*

2.3.3 Header File Details: AmbaIQParamXXX_A12_ADJ_StillLIso.c

The parameters for the still mode should be tuned for the best snapshot quality. One group of settings is defined to fit all snapshot circumstances: (1) low-ISO mode. The group has its own quality requirements and different filtering, correction, and color settings. Other settings, like AE and AWB ratios, can be shared among these modes and defined in `AmbaIQParamXXX_A12_Adj_VideoXXX.c`. (Please refer to Section 2.3.2.1 and Table 2-24)

2.3.3.1 AmbaIQParamXXX_A12_Adj_StillLISO.c : Programming Map

2.3.3.1.1 AmbaIQParamXXX_A12_ADJ_StillLIso.c > ADJ_Still_FAST_LISO_PARAM_s

The `AmbaIQParamXXX_A12_ADJ_LIso.c` parameter provides settings that take effect only when capturing in the low ISO mode.

```
typedef struct _ADJ_STILL_FAST_LISO_PARAM_s {
UINT32                VersionNum;
UINT32                ParamVersionNum;
UINT8                 NfMaxTableCount;
ADJ_FILTER_INFO_s     NormalEvImg;
ADJ_FILTER_INFO_s     FlashEvImg;
ADJ_DEF_s             Def;
ADJ_BASIC_s           Basic;

ADJ_LUT_AGC_WB_s      ChromaFilter;
DEF_SHARP_INFO_s      SharpInfo;

} ADJ_STILL_FAST_LISO_PARAM_S;
```

2.3.4 Header File Details: AmbaIQParamXXX_A12_ADJ_StillHIso.c

The parameters for the high ISO still mode should be tuned for the best snapshot quality. Other settings, like AE and AWB ratios, can be shared among these modes and defined in `AmbaIQParamXXX_A12_Adj_VideoXXX.c`. (Please refer to Section 2.3.2.1 and Table 2-24)

2.3.4.1 AmbaIQParamXXX_A12_Adj_StillHISO.c : Programming Map

- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > ADJ_FILTER_INFO_s(Section 2.3.2.1.1.4)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > ADJ_DEF_s (Section 2.3.2.1.1.5)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > ADJ_BASIC_s(Section 2.3.2.1.1.10)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > DEF_SHARP_INFO_s (Section 2.3.2.1.1.12)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > ADJ_HISO_FILTER_INFO_s(Table 2-3-2-14)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > ADJ_BASIC_s(Section 2.3.2.1.1.10)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > HIsoCdnr(Section 2.3.2.1.1.12)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > DEF_ASF_INFO_s(Section 2.3.2.1.1.12)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > DEF_SHARP_s(Section 2.3.2.1.1.12)
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > CHROMA_FILTER_COMBINE_s
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > LUMA_COMBINE_s
- AmbaIQParamXXX_A12_Adj_StillHISOXXX.c > ADJ_STILL_HISO_PARAM_s > DEF_FIR_s(Section 2.3.2.1.1.13)

2.3.4.1.1 AmbaIQParamXXX_A12_ADJ_StillHIso.c > ADJ_Still_HISO_PARAM_s

The `AmbaIQParamXXX_A12_ADJ_HIso.c` parameter provides settings that take effect only when capturing in the High ISO mode.

```
typedef struct _ADJ_STILL_HISO_PARAM_s_ {
          UINT32              VersionNum;
          UINT32              ParamVersionNum;
          UINT8               NfMaxTableCount;
          ADJ_FILTER_INFO_s   NormalEvImg;
          ADJ_FILTER_INFO_s   FlashEvImg;
          ADJ_DEF_s          Def;
          ADJ_BASIC_s        Basic;
          ADJ_LUT_AGC_WB_s   ChromaFilter;
          DEF_SHARP_INFO_s   SharpInfo;

// Start of HISO
          ADJ_HISO_FILTER_INFO_s         HIsoNormalEvImg;
          ADJ_HISO_FILTER_INFO_s         HIsoFlashEvImg;
          ADJ_BASIC_s                    HIsoBasic;
          UINT8                          HIsoCdnrEnable;
          ADJ_LUT_s                      HIsoCdnrLut[ADJ_HISO_NF_TABLE_COUNT];
          DEF_ASF_INFO_s                 HIsoAsf;
          DEF_ASF_INFO_s                 HIsoHighAsf;
          DEF_ASF_INFO_s                 HIsoMed1Asf;
          DEF_ASF_INFO_s                 HIsoMed2Asf;
          DEF_ASF_INFO_s                 HIsoLowAsf;
          DEF_SHARP_s                    HIsoHighSharp;
          DEF_SHARP_s                    HIsoMedSharp;
          DEF_SHARP_s                    HIsoLiSharp;
          DEF_ASF_INFO_s                 HIsoChromaAsf;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterPre;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterHigh;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterMed;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterLow;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterVLow;
          ADJ_LUT_AGC_WB_s               HIsoChromaFilterLowAndVLow;
          CHROMA_FILTER_COMBINE_s        HIsoChromaFilterMedCombine;
          CHROMA_FILTER_COMBINE_s        HIsoChromaFilterLowCombine;
          CHROMA_FILTER_COMBINE_s        HIsoChromaFilterVLowCombine;
          LUMA_COMBINE_s                 HIsoLumaFilterCombine;
          LUMA_COMBINE_s                 HIsoLowAsfCombine;
          CHROMA_FILTER_COMBINE_s        HIsoLiCombine;
          UINT8                          HIsoLiLumaMidHightFreqRcvrEnable;
          DEF_FIR_s                      HIsoLiLumaMidHightFreqRcvr;
          UINT8                          HIsoLi2ndBlendEnable;
          ADJ_LUT_s                      HIsoLi2ndBlend[ADJ_HISO_NF_TABLE_COUNT];
          DEF_ASF_INFO_s                 Li2ndAsf;
          DEF_SHARP_s                    Li2ndSharp;
} ADJ_STILL_HISO_PARAM_s;
```

2.3.5 Header File Details: AmbaIQParamXXX_A12_ImageParam.c

The header file `AmbaIQParamXXX_ImageParams.c` provides parameters to set the initial black level, bad pixel correction, **wb_gain**, gamma, **rgb_to_yuv** matrix, CFA noise filter, and **luma_sharpen** filters, and the AE/AWB/AF statistics settings.

2.3.5.1 AmbaIQParamXXX_A12_ImageParam > IMG_PARAM_s

```
typedef struct _IMG_PARAM_s_ {
UINT32                                VersionNum;
AMBA_DSP_IMG_BLACK_CORRECTION_s       BlackCorrVideo;
AMBA_DSP_IMG_BLACK_CORRECTION_s       BlackCorrStill;
AMBA_DSP_IMG_DBP_CORRECTION_s         BadCorrVideo;
AMBA_DSP_IMG_DBP_CORRECTION_s         BadCorrStill;
AMBA_DSP_IMG_CFA_LEAKAGE_FILTER_s     CfaLeakageFilterVideo;
AMBA_DSP_IMG_CFA_LEAKAGE_FILTER_s     CfaLeakageFilterStill;
AMBA_DSP_IMG_CFA_NOISE_FILTER_s       CfaNoiseFilterVideo;
AMBA_DSP_IMG_CFA_NOISE_FILTER_s       CfaNoiseFilterStill;
AMBA_DSP_IMG_GBGR_MISMATCH_s          GrGbMismatchVideo;
AMBA_DSP_IMG_GBGR_MISMATCH_s          GrGbMismatchStill;
AMBA_DSP_IMG_DEMOSAIC_s               DemosaicVideo;
AMBA_DSP_IMG_DEMOSAIC_s               DemosaicStill;

UINT8                                 AntiAliasingEnableVideo;
UINT8                                 AntiAliasingEnableStill;
AMBA_DSP_IMG_AAA_STAT_INFO_s          AaaStatisticsInfo;
AMBA_DSP_IMG_WB_GAIN_s                WbGainVideo;
AMBA_DSP_IMG_WB_GAIN_s                WbGainStill;
AMBA_DSP_IMG_LOCAL_EXPOSURE_s         LocalExposureVideo;
AMBA_DSP_IMG_LOCAL_EXPOSURE_s         LocalExposureStill;

AMBA_DSP_IMG_COLOR_CORRECTION_s       ColorCorrVideo;
AMBA_DSP_IMG_COLOR_CORRECTION_s       ColorCorrStill;

AMBA_DSP_IMG_TONE_CURVE_s             ToneCurveVideo;
AMBA_DSP_IMG_TONE_CURVE_s             ToneCurveStill;

AMBA_DSP_IMG_RGB_TO_YUV_s             RgbYuvMatrixVideoTv;
AMBA_DSP_IMG_RGB_TO_YUV_s             RgbYuvMatrixVideoPc;
AMBA_DSP_IMG_RGB_TO_YUV_s             RgbYuvMatrixStill;

AMBA_DSP_IMG_CHROMA_SCALE_s           ChromaScaleVideo;
AMBA_DSP_IMG_CHROMA_SCALE_s           ChromaScaleStill;
AMBA_DSP_IMG_CHROMA_MEDIAN_FILTER_s   ChromaMedianFilterVideo;
AMBA_DSP_IMG_CHROMA_MEDIAN_FILTER_s   ChromaMedianFilterStill;

AMBA_DSP_IMG_CHROMA_FILTER_s          ChromaFilterVideo;
AMBA_DSP_IMG_CHROMA_FILTER_s          ChromaFilterStill;

UINT16                                VideoGammaCurve[TONE_CURVE_SIZE];
UINT16                                StillGammaCurve[TONE_CURVE_SIZE];
AMBA_DSP_IMG_DGAIN_SATURATION_s       DgainSaturation;

AMBA_DSP_IMG_CDNR_INFO_s              CdnrVideo;
AMBA_DSP_IMG_CDNR_INFO_s              CdnrStill;

/* Warp  and MCTF  related filters */
AMBA_DSP_IMG_VIDEO_MCTF_INFO_s        MctfInfoVideo;
} IMG_PARAM_s;
```

The field definitions are as follows:

Type	Field	Description
UINT32	**VersionNum**	Version number of this structure
structure	**BlackCorrVideo**	Video/Still Black Level initial settings
structure	**BlackCorrStill**	
structure	**BadCorrVideo**	Video/Still Dynamic Bad Pixel Correction initial settings
structure	**BadCorrStill**	
structure	**CfaLeakageFilterVideo**	Video/Still color filter array (CFA) Leakage Filter initial settings
structure	**CfaLeakageFilterStill**	
structure	**CfaNoiseFilterVideo**	Video/Still CFA Noise Filter initial settings
structure	**CfaNoiseFilterStill**	
structure	**GrGbMismatchVideo**	Video/Still GrGb Mismatch Filter initial settings
structure	**GrGbMismatchStill**	
structure	**DemosaicVideo**	Video/Still Demosaic Filter initial settings
structure	**DemosaicStill**	
Uint8	**AntiAliasingEnableVideo**	Video/Still AntiAliasing Filter initial settings
Uint8	**AntiAliasingEnableStill**	
AMBA_DSP_IMG_AAA_STAT_INFO_s	**AaaStatisticsInfo**	AE/AWB statistics settings. The user should not modify it unless needed.
structure	**WbGainVideo**	Video/Still WB Gain initial settings
structure	**WbGainStill**	
structure	**LocalExposureVideo**	Video/Still Local Exposure initial settings
structure	**LocalExposureStill**	
structure	**ColorCorrVideo**	Color Correction settings. The user should not modify it.
structure	**ColorCorrStill**	
structure	**ToneCurveVideo**	Video/Still output table initial settings These are the output tables for gamma-ratio = 128.
structure	**ToneCurveStill**	
structure	**RgbYuvMatrixVideoTv**	VideoTv/VideoPc/Still RgbYuv Matrix settings
structure	**RgbYuvMatrixVideoPc**	
structure	**RgbYuvMatrixStill**	
structure	**ChromaScaleVideo**	Video/Still ChromaScale Filter intial settings These are the curves for chroma-scale ratio = 128.
structure	**ChromaScaleStill**	
structure	**ChromaMedianFilterVideo**	Video/Still Chroma Median Filter intial settings
structure	**ChromaMedianFilterStill**	
structure	**ChromaFilterVideo**	Video/Still Chroma Noise Filter intial settings
structure	**ChromaFilterStill**	
UINT16	**VideoGammaCurve[]**	TBD.
UINT16	**StillGammaCurve[]**	
structure	**DGainSaturation**	Specifies the allowed maximum digital gain for various preset modes and saturation levels of different color channels. The valid range is 15 bits. If (raw_pixel_value * dgain) > dgain_saturation, the output after the dgain stage will be clamped to the **dgain_saturation** value.
structure	**CdnrVideo**	TBD
structure	**CdnrStill**	
structure	**MctfInfoVideo**	MCTF Filter intial settings

Table 2-66. Field Definition for the `AmbaIQParamXXX_A12_ImageParam.c` *Structure* ***IMG_PARAM_s()****.*

2.3.6 Header File Details: AmbaIQParamXXX_A12_ScXXXParam.c

Type	Field	Description
AmpIQParamXXX_A12_ScSet01Param.c	**SCENE_DATA_s**	SenceDataS01xxxA12[8] (OFF, FLASH, TV_OFF, AV_OFF, SV_OFF, TV_ONLY, AV_ONLY, SV_ONLY) Please refer to Section 2.3.6.1.
AmpIQParamXXX_A12_ScSet02Param.c	**SCENE_DATA_s**	SenceDataS02xxxA12 [8] (NIGHT, NIGHT_PORTRAIT, SPORTS, LANDSCAPE, PORTRAIT, SUNSET, SAND_SNOW, FLOWER) Please refer to Section 2.3.6.1.
AmpIQParamXXX_A12_ScSet03Param.c	**SCENE_DATA_s**	SenceDataS03xxxA12 [8] (FIRE_WORK, WATER, BACK_LIGHT, BACK_LIGHT_PORTRAIT, TRIPOD, BLUE_SKY, MACRO, MACRO_TEXT) Please refer to Section 2.3.6.1.
AmpIQParamXXX_A12_ScSet04Param.c	**SCENE_DATA_s**	SenceDataS04xxxA12 [8] (ARENA, D_LIGHTING, MUSEUM, BEACH, CHILDREN, PARTY, FISHEYE, INDOOR) Please refer to Section 2.3.6.1.
AmpIQParamXXX_A12_ScSet05Param.c	**SCENE_DATA_s**	SenceDataS05xxxA12 [8] (THROUGH_GLASS, PANNING, PHOTO_FRAME, LOMO, SELF_PORTAIT, CAR_DV) Please refer to Section 2.3.6.1.

*Table 2-67. Header `AmbaIQPaamXXX_A12_ScXXXParam.c` Settings to scene modes**()**.*

2.3.6.1 AmbaIQParamXXX_A12_ScXXXParam.c : Programming Map

- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_DEF_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_VIDEO_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s > LUT_CONTROL_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s > AE_LUT _s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AWB_s
- AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_ADJ_s

2.3.6.1.1 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_DEF_s

The `AmbaIQParamXXX_A12_ScXXXParam.c` data structure **SCENE_DEF_s** is shown below:

```
typedef struct_SCENE_DEF_s_ {
  UINT8 ColorTable;
  UINT8 DigitalEffect;
} SCENE_DEF_s;
```

The field definitions are as follows:

Type	Field	Description
UINT8	**ColorTable**	Specifies the number of Color Correction (CC) matrix table to be used for this scene mode. The default setting is **SYSTEM_DEFAULT**.
UINT8	**DigitalEffect**	Specifies the settings of the digital effect to be used for this scene mode. The default setting is **SYSTEM_DEFAULT**.

*Table 2-68. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` structure **SCENE_DEF_s()**.*

2.3.6.1.2 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s

The `AmbaIQParamXXX_A12_ScXXXParam.c` data structure **SCENE_AE_s** is shown below:

```
typedef struct _SCENE_AE_s_ {
  SCENE_VIDEO_s     Video;
  SCENE_VIDEO_s     Photo;
  SCENE_STILL_s     Still;
} SCENE_AE_s
```

The field definitions are as follows:

Type	Field	Description
structure	**Video**	Specifies the automatic exposure (AE) settings of video mode in this scene mode. Please refer to **SCENE_VIDEO_s** in Section 2.3.6.1.3 below.
structure	**Photo**	Specifies the AE settings of photo-preview mode in this scene mode. Please refer to **SCENE_VIDEO_s** in Section 2.3.6.1.3 below.
structure	**Still**	Specifies the AE settings of the still mode in this scene mode. Please refer to **SCENE_STILL_s** in Section 2.3.6.1.4 below.

*Table 2-69. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` Structure **SCENE_DEF_s** > **SCENE_AE_s()**.*

2.3.6.1.3 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_VIDEO_s

The AmbaIQParamXXX_A12_ScXXXParam.c data structure SCENE_VIDEO_s is shown below:

```
typedef struct _SCENE_VIDEO_s_ {
  UINT8       AvTvMode;
  UINT16      DefExp[4];
  UINT8       SlowShutter;
  UINT8       Fps60SlowshutterFps;    //60fps,50fps, 30fps,25fps,
                                  15fps,12fps,7fps,6fps
  UINT8       Fps30SlowshutterFps;    //60fps,50fps, 30fps,25fps,
                                  15fps,12fps,7fps,6fps
  UINT8       Flash;
} SCENE_VIDEO_s;
```

The field definitions are as follows:

Type	Field	Description
UINT8	**AvTvMode**	AE mode for this scene, please refer to `AmbaImg_AaaDef.h`. In different AvTvModes; the AGC, Shutter, and IRIS adjustment will be different while performing AE, and this parameter is assigned according to the variation of the current scene.
UINT16	**DefExp[4]**	TBD.
UINT8	**SlowShutter**	DISABLE ENABLE SYSTEM_DEFAULT Enable/Disable slow shutter for this scene. When this parameter is set at **SYSTEM_DEFAULT**, it applies the slow-shutter setting in the file `AmbaIQParamXXX_DefaultParams.c`.
UINT8	**Fps60SlowshutterFps**	Specifies the slow shutter FPS for 60-fps of this scene.
UINT8	**Fps30SlowshutterFps**	Specifies the slow shutter FPS for 30-fps of this scene.
UINT8	**Flash**	**FLASH_ALWAYS_OFF** **FLASH_AUTO** **FLASH_ALWAYS_ON** **SYSTEM_DEFAULT** Specifies the flash function.

*Table 2-70. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` Structure **SCENE_DEF_s SCENE_AE_s > SCENE_VIDEO_s ()**.*

2.3.6.1.4 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s

The AmbaIQParamXXX_A12_ScXXXParam.c data structure **SCENE_STILL_s** is shown below:

```
typedef struct _SCENE_STILL_s_ {
  AE_EXPO_CONTROL_s       ExpoControl;
  UINT8                   NightShot;
  UINT8                   Is;
  UINT8                   Flash;
  UINT8                   FlashType;
} SCENE_STILL_s;
```

The field definitions are as follows:

Type	Field	Description
structure	**ExpoControl**	Specifies the exposure settings of the still mode in this scene mode. Please refer to **AE_EXPO_CONTROL_s** in Section 2.3.6.1.5.
UINT8	**NightShot**	Specifies the NIGHT-SHOT function: ENABLE DISABLE **SYSTEM_DEFAULT**
UINT8	**Is**	Specifies the STILL-IS function: ENABLE DISABLE **SYSTEM_DEFAULT**
UINT8	**Flash**	Specifies the flash function: **FLASH_ALWAYS_OFF** **FLASH_AUTO** **FLASH_ALWAYS_ON** **SYSTEM_DEFAULT**
UINT8	**FlashType**	Specifies the flash type: **FLASH_ALWAYS_OFF** **AE_FLASH_NORMAL** **AE_FLASH_SLOW** **SYSTEM_DEFAULT**

*Table 2-71. Field Definition for the AmbaIQParamXXX_A12_ScXXXParam.c Structure **SCENE_DEF_s SCENE_AE_s > SCENE_STILL_s()**.*

2.3.6.1.5 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s

The `AmbaIQParamXXX_A12_ScXXXParam.c` data structure **AE_EXPO_CONTROL_s** is shown below:

```
typedef struct _AE_EXPO_CONTROL_s_ {
  UINT32       MinIsoValue;
  UINT32       MaxIsoValue;
  UINT32       MaxIsoValueHiso;
  LUT_CONTROL_s     ZoomVector;
  UINT8        TableCount;
  UINT8        ExpoLutChk;
  AE_LUT_s     ExpoLut[24];
  LUT_CONTROL_s     MotionIsoRatio;
} AE_EXPO_CONTROL_s;
```

The field definitions are as follows:

Type	Field	Description
UINT32	**MinIsoValue**	Specifies the minimum ISO value of auto-ISO still mode for this scene mode.
UINT32	**MaxIsoValue**	Specifies the maximum ISO value of auto-ISO still mode for this scene mode.
UINT32	**MaxIsoValueHiso**	Specifies the maximum ISO value of high-ISO still capture for this scene mode.
structure	**ZoomVector**	Please refer to **Zoom_Vector** in Section 2.3.6.1.6.
UINT8	**TableCount**	Specifies the number of groups of **shutter_index**, IRIS, and ISO values for still capture. This number is dependent on the range of output specified by values of str from **lut_control**. (Up to 5).
UINT8	**ExpoLutChk**	**EXPO_CHK_SHUTTER**: Decides the exposure by the shutter index of the table ExpoLut[24]. **EXPO_CHK_ISO**: Decides the exposure by the ISO value of the table ExpoLut[24].
structure	**ExpoLut[24]**	Please refer to **AE_LUT_s** in Section 2.3.6.1.7.
structure	**MotionIsoRatio**	Equal to a look-up table. The index of the table is the current motion vector. The outputs from motion_vector will be regarded as the input of **LUT_CONTROL_s** and the final output is the indexing of the **shutter_index** and the ISO value for still capture

*Table 2-72. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` Structure **SCENE_DEF_s SCENE_AE_s > SCENE_VIDEO_s_ > AE_EXPO_CONTROL_s()**.*

2.3.6.1.6 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s > LUT_CONTROL_s

The parameter equals to a look-up table, and the indexing of the table is the currently optical zoom step. The outputs from the ZoomVector will be regarded as the inputs for **LUT_CONTROL_s**. The final output is the indexing of **shutter_index** and ISO value for still capture.

```
typedef struct _LUT_CONTROL_s_ {
  INT16       Start;
  INT16       End;
  STR_LUT_s   Lut;
} LUT_CONTROL_s;
```

The field definitions are as follows:

Type	Field	Description
INT16	**Start**	Specifies the start value of the optical zoom step. (For indexing)
INT16	**End**	Specifies the end value of the optical zoom step. (For indexing)
structure	**Lut**	Specifies the corresponding output ISO values by using look-up tables for the input optical zoom step.

*Table 2-73. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` structure **SCENE_DEF_s SCENE_AE_s > SCENE_VIDEO_s_ > AE_EXPO_CONTROL_s > LUT_CONTROL_s()**.*

2.3.6.1.7 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AE_s > SCENE_STILL_s > AE_EXPO_CONTROL_s > AE_LUT _s

The parameter **ExpoLut[24]** specifies the shutter_index, IRIS, and ISO values of still capture for each luminance level. Each level contains shutter_index, IRIS, and ISO values of several groups. (The number of groups are specified by **table_count**) The details of the structure and the example table are shown below:

//vector:0			128			256			
//SHUTTER	ISO	IRIS	SHUTTER	ISO	IRIS	SHUTTER	ISO	IRIS	
{244,	800	1	244,	800,	1,	244,	800,	1},	//LV0
{372,	800	1	244,	800,	1,	244,	800,	1},	//LV1
{500,	800	1	372,	800,	1,	244,	800,	1},	//LV2
{ 2036,	AE_ISO_ MIN,	82,	2036,	AE_ISO_ MIN,	1,	2036,	AE_ ISO_ MIN,	1},	//LV20

*Table 2-74. **ExpoLut[24]**.*

2.3.6.1.8 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_AWB_s

The `AmbaIQParamXXX_A12_ScXXXParam.c` data structure **SceneAWB_s** is shown below:

```
typedef struct _SCENE_AWB_s_ {
  UINT8       MenuMode;
  UINT8       MenuModeType;
} SCENE_AWB_s;
```

The field definitions are:

Type	Field	Description
UINT8	**MenuMode**	This index is used to set the AWB mode, and it could be **SYSTEM_DEFAULT**, **WB_AUTOMATIC**, **INCANDESCENT**, **SUNNY**, etc. Please refer to the related files.
UINT8	**MenuModeType**	Specifies the method of generating the AWB gain when the user enters the menu mode, and it could be **WB_MENU_REGION**, **WB_MENU_FIX** and **SYSTEM_DEFAULT**.

*Table 2-75. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` structure **SCENE_AWB_s()**.*

2.3.6.1.9 AmbaIQParamXXX_A12_ScXXXParam.c > SCENE_DATA_s > SCENE_ADJ_s

This structure is used to fine tune the IQ (including low/high temperature R and B ratio, AE target, saturation, gamma- ratio, local-exposure-ratio, and auto-knee) for this scene. The definitions for **SCENE_ADJ_s** are:

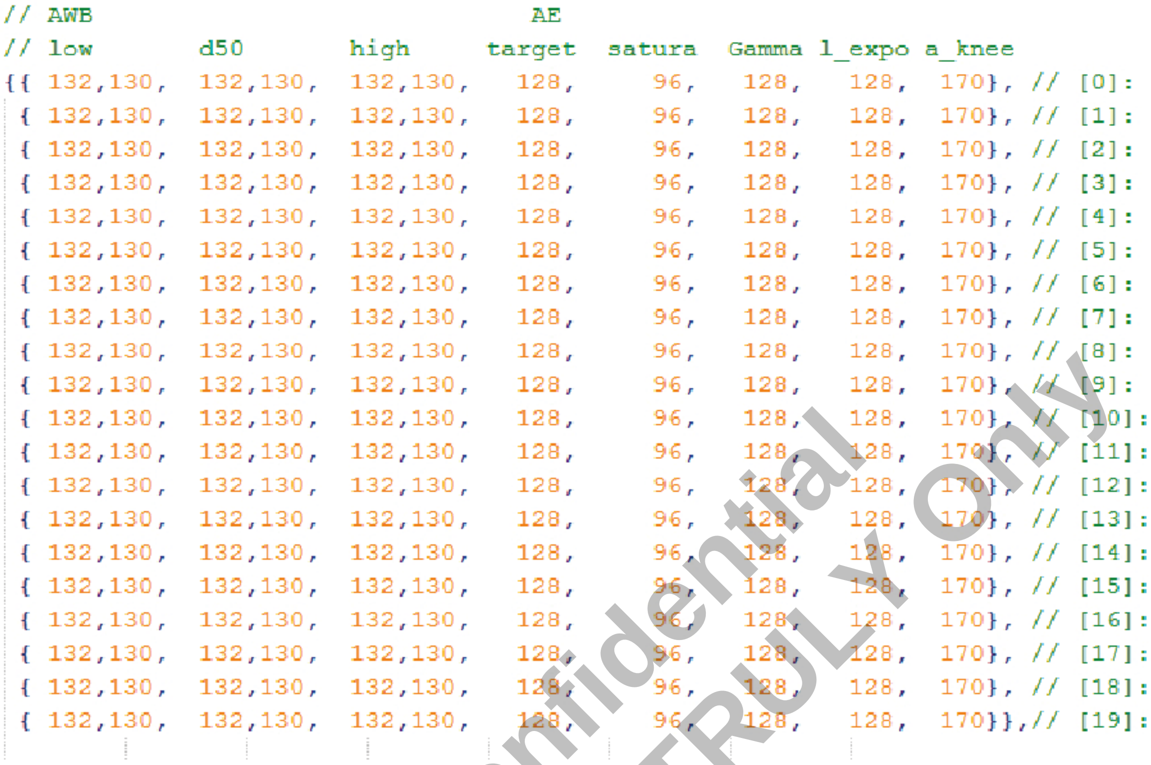
```
typedef struct _SCENE_ADJ_s_ {
  UINT16      LightCondition;
  ADJ_LUT_s   VideoTable[25];
  ADJ_LUT_s   StillTable[25];
} SCENE_ADJ_s;
```

The field definitions are as follows:

Type	Field	Description
UINT16	**LightCondition**	**LIGHT_CONDITION_OFF**: Do not apply the histogram-base adjustment. 0, 1, 2…: Apply the histogram-base adjustment according to the Lighting-Condition setting in the file `AmbaIQParamXXX_A12_DefaultParams.c`. Please refer Section 2.3.1.1.1.19. According to this parameter, the Video/Still Table should be different, please refer **ex-1** and **ex-2**.
structure	**VideoTable[25]**	Each **VideoTable/StillTable** is the fine-tuned ratio for the video/still version of this scene.
structure	**StillTable[25]**	

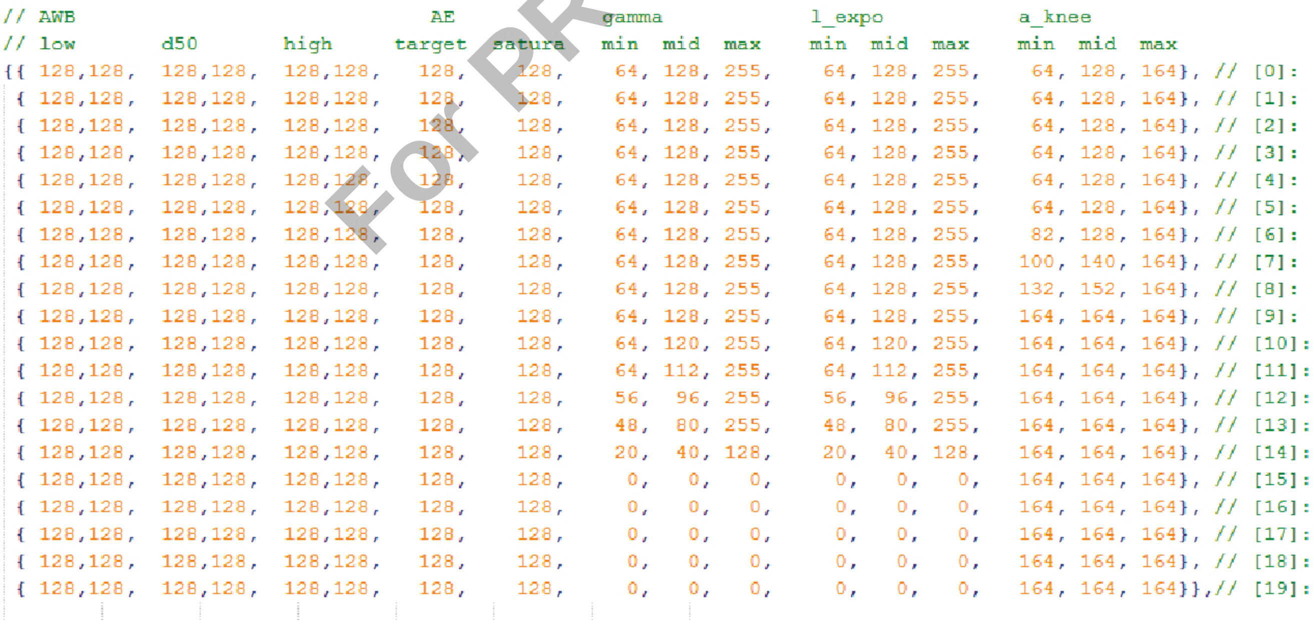
*Table 2-76. Field Definition for the `AmbaIQParamXXX_A12_ScXXXParam.c` Structure **SCENE_ADJ_s()**.*

A **video_table/still_table** specifies up to 25 ev-index levels for one scene. These ratios are for related IQ tuning items in app `AmbaIQParamXXX_A12_Adj_XXX.c`, including R and B ratio for low/D50/high color temperature (2x3), AE target (1), saturation control (1), gamma-ratio (3), local-exposure-ratio (3) and auto_knee (3). The adjustment parameters from `AmbaIQParamXXX_A12_Adj_XXX.c` will cascade these values for optimal performance.



```
// AWB                                  AE
// low       d50       high       target  satura  Gamma l_expo a_knee
{{ 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [0]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [1]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [2]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [3]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [4]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [5]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [6]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [7]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [8]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [9]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [10]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [11]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [12]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [13]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [14]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [15]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [16]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [17]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}, // [18]:
 { 132,130,  132,130,  132,130,    128,     96,    128,    128,   170}},// [19]:
```

Table 2-77. ***ex-1****.*



```
// AWB                                  AE          gamma            l_expo            a_knee
// low        d50        high        target  satura  min  mid  max   min  mid  max    min  mid  max
{{ 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [0]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [1]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [2]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [3]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [4]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    64, 128, 164}, // [5]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,    82, 128, 164}, // [6]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,   100, 140, 164}, // [7]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,   132, 152, 164}, // [8]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 128, 255,   64, 128, 255,   164, 164, 164}, // [9]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 120, 255,   64, 120, 255,   164, 164, 164}, // [10]:
 { 128,128,  128,128,  128,128,    128,   128,    64, 112, 255,   64, 112, 255,   164, 164, 164}, // [11]:
 { 128,128,  128,128,  128,128,    128,   128,    56,  96, 255,   56,  96, 255,   164, 164, 164}, // [12]:
 { 128,128,  128,128,  128,128,    128,   128,    48,  80, 255,   48,  80, 255,   164, 164, 164}, // [13]:
 { 128,128,  128,128,  128,128,    128,   128,    20,  40, 128,   20,  40, 128,   164, 164, 164}, // [14]:
 { 128,128,  128,128,  128,128,    128,   128,     0,   0,   0,    0,   0,   0,   164, 164, 164}, // [15]:
 { 128,128,  128,128,  128,128,    128,   128,     0,   0,   0,    0,   0,   0,   164, 164, 164}, // [16]:
 { 128,128,  128,128,  128,128,    128,   128,     0,   0,   0,    0,   0,   0,   164, 164, 164}, // [17]:
 { 128,128,  128,128,  128,128,    128,   128,     0,   0,   0,    0,   0,   0,   164, 164, 164}, // [18]:
 { 128,128,  128,128,  128,128,    128,   128,     0,   0,   0,    0,   0,   0,   164, 164, 164}},// [19]:
```

Table 2-78. ***ex-2****.*

Field	Description
Low temp R ratio	These 6 values are adjustments of the AWB for this scene. The units are 1/128.
Low temp B ratio	
D50 R ratio	
D50 B ratio	
High temp R ratio	
High temp B ratio	
AE target ratio	This adjusts the AE target ratio defined in `AmbaIQParamXXX_A12_Adj_XXX.c` and the unit value is 128 .
Saturation	This value adjusts the chroma-scale ratio defined in `AmbaIQParamXXX_A12_Adj_XXX.c` and the scale is 128 for the unit.
Gamma-ratio	This adjusts the gamma ratio defined in AmbaIQParamXXX_A12_Adj_XXX.c and the scale is 128 for the unit.
Local-Exposure ratio	This value adjusts the local-exposure ratio defined in `AmbaIQParamXXX_A12_Adj_XXX.c` and the scale is 128 for the unit.
Auto-knee	This value will modify the luminance weighting style of AE. A value of 128 implies that no effect occurs. If it is greater than 128 (maximum 255), more details will be revealed in the area that is highlighted. Values less than 128 (minimal: 0) are set for revealing the details in the dark area.

Table 2-79.   Detail from `AmbaIQParamXXX_A12_ScXXXParam.c` *structure* ***scene_adj_t()***.

The settings of gamma-ratio, local-exposure-ratio, and auto-knee have similar structure (Please refer to Section 2.3.1.1.1.19). They are separated into different parts (min, mid, and max) and defined in the scene control parameter settings of `AmbaIQParamXXX_A12_DefaultParams.c`.

2.3.7   Header File Details: AmbaIQParamXXX_A12_DeXXXParam.c

There are two kinds of parameters for IQ settings of digital effect, as shown in Section Table 2-80. The header file AmbaIQ`ParamXXX_A12DeVideo/StillParam.c` contains **DeVideo/StillParamXXXA12**, which is used for the IQ settings of video/still digital effects.

Header File	Structure	Description
AmbaIQParamXXX_A12DeVideo/ StillParam.c	**De_Param_s**	DeVideo/StillParam_xxxA12. Please refer to Section 2.3.7.1 below for definition.

Table 2-80.   Header `AmbaIQParamXXX_A12_DeXXXParam.c` *Settings to Video and Still Digital Effects**()**.*

2.3.7.1 AmbaIQParamXXX_A12_DEXXXParam.c : Programming Map

- AmbaIQParamXXX_A12_DeXXXParam.c > DE_PARAM_s > DE_SETTING_s

2.3.7.1.1 AmbaIQParamXXX_A12_DeXXXParam.c > DE_PARAM_s

```
typedef struct _DE_PARAM_s_ {
    UINT32          VersionNum;
    UINT32          ParamVersionNum;
    DE_SETTING_s    DeInfo[24];

    TONE_CURVE_s  ToneCurve[6];
    UINT16          Vignette[6][1089];
} DE_PARAM_s;
```

The field definitions are:

Type	Field	Description
UINT32	**VersionNum**	Version number
UINT 32	**ParamVersionNum**	Version number of these tuning parameters
structure	**DeInfo[24]**	Defines the AE target, cc matrix, rgb-to-yuv matrix offset, tone curve, CFA noise filter, sharpening filter, and IQ settings for this digital effect. Please refer to **DE_SETTING_s** in Section 2.3.7.1.1.1 below.
structure	**ToneCurve [6]**	Defines the tone curve table to be used for this digital effect.
UINT 16	**Vignette [6][1089]**	Defines the vignette table to be used for this digital effect.

*Table 2-81. Field Definition for the `AmbaIQParamXXX_A12_DeXXXParam.c` Structure **DE_PARAM_s()**.*

2.3.7.1.1.1 AmbaIQParamXXX_A12_DeXXXParam.c > DE_PARAM_s > DE_SETTING_s

The `AmbaIQParamXXX_A12DeVideo/StillParam.c` data structure **DE_SETTING_s** defines the AE target, cc matrix, rgb-to-yuv matrix offset, tone curve, CFA noise filter, sharpening filter, and IQ settings for this digital effect.

```
typedef struct _DE_SETTING_s_ {
  UINT16              DeMode;
  UINT16              CfaStr;
  UINT16              SharpStr;
  UINT16              LumaSmoothSr;

  UINT8               CcChangeEnable;
  UINT16              Cc3dNo;
  RGB_TORGB_INFO      Cc_Matrix;
  INT16               Rgb2Yuv[9];
  INT16               YuvOffset[3];
  UINT16              ToneCurveNo;
  UINT16              AeTargetRatio;
  UINT16              VignetteNo;
  UINT16              WarpNo;
} SCENE_ADJ_s;
```

The field definitions are as follows:

Type	Field	Description
UINT 16	**DeMode**	Defines the digital effect.
UINT 16	**CfaStr**	The strength value would reduce the CFA noise filter and sharpening filter if the value is less than 64. A value greater than 64 would enhance/enlarge the strength for each filter.
UINT 16	**SharpStr**	
UINT 16	**LumaSmoothStr**	TBD.
UINT 8	**CcChangeEnable**	ENABLE DISABLE Enable/disable to change the CC matrix table.
UINT 16	**Cc3dNo**	Specifies the number of CC matrix table to be used for this digital effect.
structure	**CcMatrix**	Reserved
INT16	**Rgb2Yuv [9]**	Specifies the 3X3 matrix to change the default RGB-to-YUV matrix. (matrix multiplication).
INT16	**YuvOffset [3]**	Specifies the YUV offset to change the default YUV offset.
UINT 16	**ToneCurveNo**	Specifies the number of tone curve tables to be used for this digital effect.
UINT 16	**AeTargetRatio**	The ratio would reduce the AE target, if the value is less than 1024. A ratio greater than 1024 would enhance/enlarge the AE target.
UINT 16	**VignetteNo**	Specifies the number of vignette tables to be used for this digital effect.
UINT 16	**WarpNo**	Specifies the number of warp tables to be used for this digital effect.

Table 2-82. Field Definition for the `AmbaIQParamXXX_A12_DeXXXParam.c` *structure* ***DE_SETTING_s()****.*

2.4 IQ Tuning: Default Binary Color Table Files

There are several default binary files required by the IDSP. They should be placed in the directory /IQ_Parameter_Files/cc/.

Ambarella provides the following default binary color table files. Users can modify them with Ambarella color tuning tools.

Field	Description
XXX_Cc_Still0.bin XXX_Cc_Still1.bin XXX_Cc_Still2.bin XXX_Cc_Still3.bin XXX_Cc_Still4.bin	The still color table tuned for a specific sensor.
XXX_Cc_Video0.bin XXX_Cc_Video1.bin XXX_Cc_Video2.bin XXX_Cc_Video3.bin XXX_Cc_Video4.bin	The video color table tuned for a specific sensor.

Table 2-83. Image Quality Tuning Default Binary Color Table Files.

Normally, the color correction (CC) files 0~2 will be generated under different color temperature, like 2800k, 5000k, and 6500k. The CC files; CC-3 and CC-4 are used for tuning. According to the adj-color-ratio tuning strategy, the CC-3 will be set as unit CC and CC-4 will be set as a favorite CC. Under the low light condition, the user can set the color ratio to be lower than 128, then the user can get the less color corrected CC to reduce the color noise (some color noise will be introduced by the CC). For the high light or any other condition that the user is interested, the user can apply the color ratio to be higher than 128. This enables the user to get the favorite CC at that condition. For example, if user wants the blue sky to be more blue at an outdoor setting, the user can apply a favorite CC to enhance the blue color, and set the color ratio to be 255. Then, the user can see a sky that is more blue.

Appendix 1 Additional Resources

Please contact an Ambarella representative for digital copies.

Appendix 2 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, "materials") are provided on an "*as is*" basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

All products are sold subject to Ambarella's terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

Ambarella assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Ambarella components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in automotive and military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer's risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 3 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
0.1	16 September 2014	Formatted to SDK6
0.2	16 September 2015	Change title to SDK6 AN A12 IQ Parameter

Table A3-1. Revision History.