CV180X & CV181X ISP Development User Guide

Version: 1.3.2

Release date: 2023-02-06

Contents

Revision History

Revision	Date	Description
0.0.1	2020/10/24	First draft
0.0.7	2020/11/24	Removed duplicate CCM chapters and added AE examples.
0.0.8	2020/12/24	Modify the value range of SatCoringLinearTh and SatCoringLinearLmt.
0.0.9	2021/05/05	Unified file format, added 182x related instructions
1.1.1	2021/06/09	Modify some file instructions and descriptions
1.1.2	2022/01/18	Remove unnecessary comments
1.2.0	2022/08/18	Update file format, related parameters and API
1.3.0	2022/09/29	Modify individual parameter comments
1.3.1	2022/10/17	Add support for cv180x
1.3.2	2023/02/06	Add ISP_SMART_EXPOSURE_ATTR_S structure description
1.3.3	2023/04/20	Fix error format and content

1 Disclaimer

Contact Us

Address Building 1, Yard 9, FengHao East Road, Haidian District, Beijing, 100094, China

Building T10, UpperCoast Park, Huizhanwan, Zhancheng Community, Fuhai Street, Baoan District, Shenzhen, 518100, China

Phone +86-10-57590723 +86-10-57590724

Website https://www.sophgo.com/

Forum https://developer.sophgo.com/forum/index.html

2 Preface

2.1 Overview

This document mainly introduces the user interface of ISP.

It can be divided into system control, 3A and ISP modules.

The first part is system control, which explains how to control ISP middleware.

The second part is 3A, which explains how to control AE, AWB and AF.

Most of the functions in this area can be debuged in CVI PqTool.

The third part is the ISP modules, which explains how to control the black level cancellation, color correction matrix, gamma, noise reduction, sharpness and other modules.

Most of the functions in this area can be debuged in CVI PqTool.

2.2 Function Overview

The control structure of ISP is shown in Fig. 2.1 .

1. Lens - project focused light signal to the sensitive area of sensor
2. sensor - after photoelectric conversion, the original image in Bayer format is sent to ISP
3. ISP –in this process, ISP controls ISP logic, lens and sensor through firmware running on it, so as to complete automatic aperture, automatic exposure, automatic white balance and other functions. Among them, the operation of firmware is driven by the interrupt of video capture unit. PQ tools can adjust ISP' s online image quality through network port or serial port. Output YUV domain image to the back-end video capture unit.

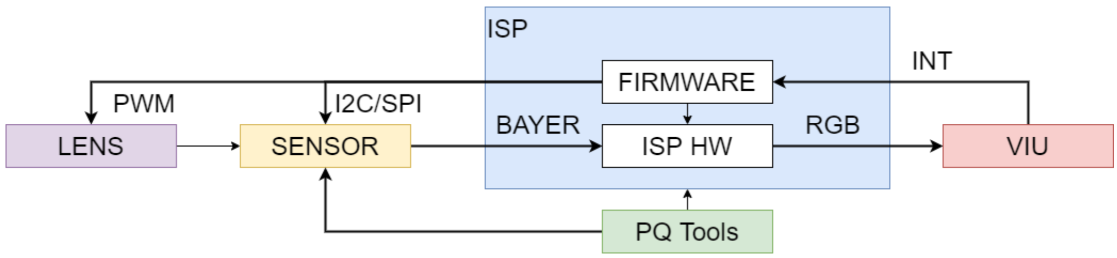


Fig. 2.1: ISP Control structure diagram

Please refer to the processor manual for ISP logic main process, specific concepts and function points.

2.3 Architecture

ISP firmware can be divided into three parts.

The first part is ISP control unit and basic algorithm library, the second part is 3A algorithm library, including AE and AWB, and the third part is the sensor library.

The software architecture is divided into three parts, and can evolve independently by registering function callbacks.

For example, developers can implement 3A functions by themselves.

As long as they implement the same interface and register, they can replace the default 3A Library of Cvi.

The ISP firmware architecture is shown in Fig. 2.2 .

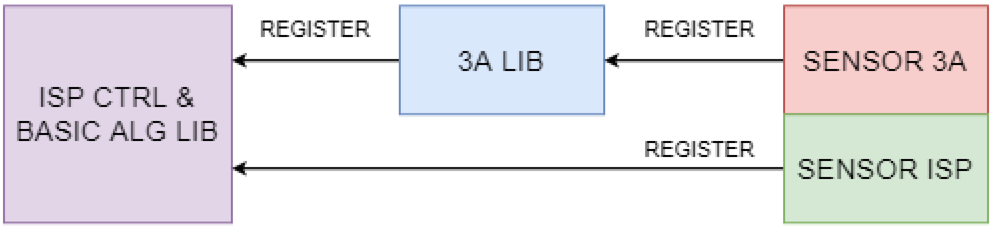


Fig. 2.2: ISP firmware Structure

Different sensors register control functions with ISP algorithm library in the form of callback functions.

When ISP control unit schedules basic algorithm library and 3A algorithm library, it will obtain initialization parameters through these callback functions and control sensor, such as adjusting analog gain, digital gain and exposure time.

2.4 Development Mode

SDK supports users to use multiple development modes.

Users use the 3A algorithm library of CVITEK.

At this time, users need to adapt different sensors according to the sensor adaptation interface given by ISP basic algorithm library and 3A algorithm library.

Each sensor corresponds to a folder, which contains two main files.

- **sensor_cmos.c** This paper mainly implements the callback function needed by ISP, including the sensor adaptation algorithm. Different sensors may be different.

- **sensor_ctrl.c** The underlying control driver of sensor mainly realizes the reading, writing and initialization of sensor.

In order to be compatible with multiple sensors at the same time, the above two file names will be added to the sensor model. For example, Sony imx307 will be named i imx307_cmos.c and imx307_ctrl.c.

Users can develop these two files according to the datasheet of sensor, and seek support from sensor manufacturer when necessary.

According to the 3A algorithm registration interface provided by ISP library, users can develop their own 3A algorithm library. At this time, users need to adapt different sensors according to the sensor adaptation interface given by ISP basic algorithm library and user 3A algorithm library. Users can use part of the CVITEK 3A algorithm library and part of their own 3A algorithm library. For example, AE uses CVITEK AE algorithm library libae.a, AWB uses its own libawb.a algorithm library.

2.5 Internal Process

The internal process of firmware is divided into two parts, as shown in Fig. 2.3 . The first part is the initialization task, which mainly completes the initialization of ISP control unit, ISP basic algorithm library and 3A algorithm library, including calling the callback of sensor to obtain the initialization parameters of sensor differentiation. The other part is the dynamic adjustment process. In this process, ISP control unit in firmware schedules ISP basic algorithm library and 3A algorithm library, calculates and controls in real time.

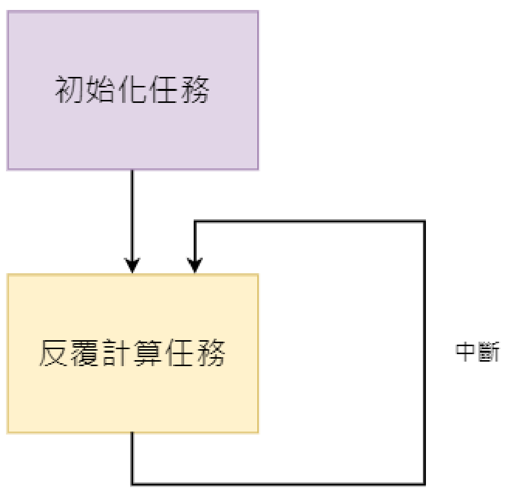


Fig. 2.3: ISP firmware Internal Process

The software structure of firmware is shown in Fig. 2.4 .

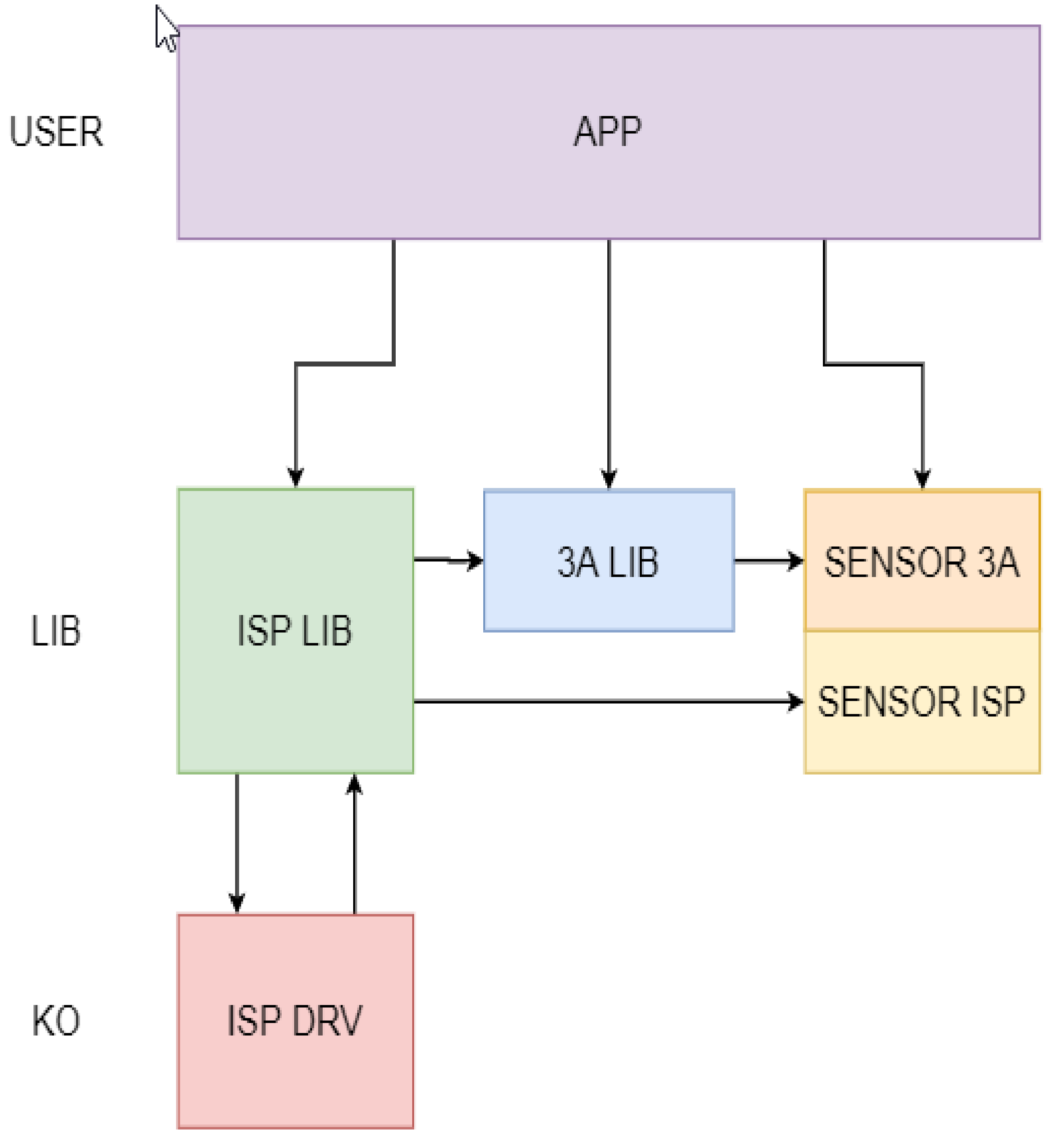


Fig. 2.4: ISP firmware Software Structure

2.6 Software Process

The software usage process is shown in Fig. 2.5 .

PQ Tools is mainly used to adjust the dynamic image quality on PC, which can adjust many factors that affect the image quality, such as denoising intensity, color conversion matrix, saturation and so on.

After debugging the image effect, users can save the configuration parameters by using the configuration file saving function provided by PQ tools. At the next startup, the system can use the configuration file loading function provided by PQ tools to load the adjusted image parameters.

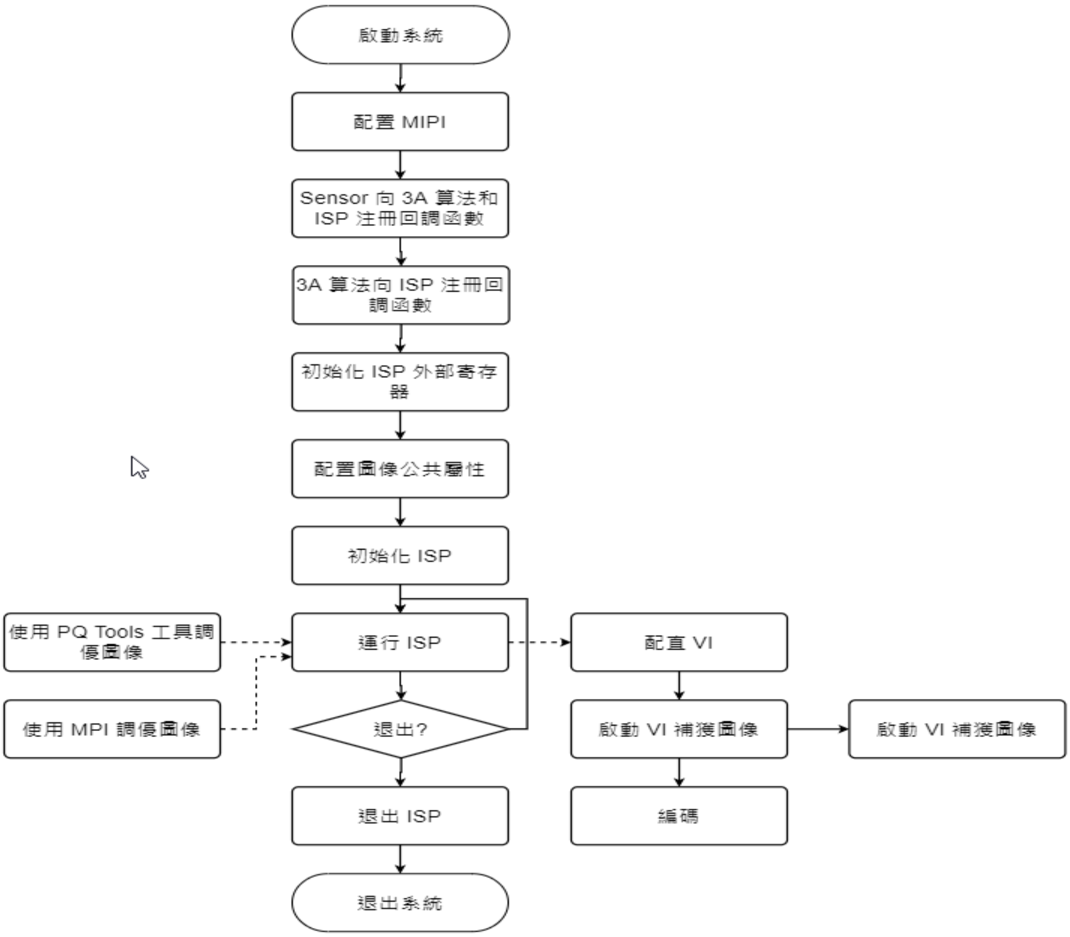


Fig. 2.5: ISP software usage process

3 System Control

3.1 Function Overview

The system control part includes the configuration of ISP public properties, initialization of ISP firmware, running of ISP firmware, exiting of ISP firmware, setting of ISP modules and other functions.

3.2 API Reference

The interface in this document does not support multi process unless otherwise specified.

- *CVI_ISP_MemInit*: Initialize the ISP external register.
- *CVI_ISP_Init*: Initialize ISP firmware.
- *CVI_ISP_Run*: Run ISP firmware.
- *CVI_ISP_RunOnce*: Run ISP firmware once.
- *CVI_ISP_Exit*: Exit ISP firmware.
- *CVI_ISP_SetPubAttr*: Set ISP public properties.
- *CVI_ISP_GetPubAttr*: Get ISP public properties.
- *CVI_ISP_SetFMWState*: Set ISP firmware status.
- *CVI_ISP_GetFMWState*: Get ISP firmware status.
- *CVI_ISP_SetModuleControl*: Set the control of ISP function module.
- *CVI_ISP_GetModuleControl*: Get the control of ISP function module.
- *CVI_ISP_GetVDTimeOut*: Get ISP interrupt information.
- *CVI_ISP_SensorRegCallBack*: The callback interface of sensor registration provided by ISP.
- *CVI_ISP_SensorUnRegCallBack*: The callback interface of sensor anti registration provided by ISP.
- *CVI_ISP_AELibRegCallBack*: The callback interface of AE library registration provided by ISP.

- *CVI_ISP_AELibUnRegCallBack*: The callback interface of AE library anti registration provided by ISP.
- *CVI_ISP_AWBLibRegCallBack*: The callback interface of AWB library registration provided by ISP.
- *CVI_ISP_AWBLibUnRegCallBack*: The callback interface of AWB library anti registration provided by ISP.
- *CVI_ISP_SetBindAttr*: Set the binding relationship between ISP library, 3A library and sensor.
- *CVI_ISP_GetBindAttr*: Get the binding relationship between ISP library, 3A library and sensor.
- *CVI_ISP_SetCtrlParam*: Set control parameters of ISP.
- *CVI_ISP_GetCtrlParam*: Get control parameters of ISP.
- *CVI_ISP_SetModParam*: Set ISP module parameters.
- *CVI_ISP_GetModParam*: Get ISP module parameters.
- *CVI_BIN_SetBinName* : Set the path and file name of PQBIN storage
- *CVI_BIN_GetBinName* : Get the path and file name of PQBIN storage
- *CVI_BIN_GetBinExtraAttr* : Get bin header data information.
- *CVI_BIN_GetBinTotalLen* : Get the total length of bin data.
- *CVI_BIN_ExportBinData* : Save parameters into PQBin file.
- *CVI_BIN_ImportBinData* : Parse the data of all modules from PQBin.
- *CVI_BIN_SaveParamToBin* : Save parameters to PQBin file.
- *CVI_BIN_LoadParamFromBin* : Parse all module data from PQBin.
- *CVI_ISP_IrAutoRunOnce* : Run the infrared automatic switching function.
- *CVI_ISP_SetSmartInfo* : Set intelligent identification area information.
- *CVI_ISP_GetSmartInfo* : Get smart identification area information.

3.2.1 CVI_ISP_MemInit

【Description】

Initialize the ISP external register.

【Syntax】

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

【Parameter】

Parameter	Description
ViPipe	ViPipe number

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_Exit*

3.2.2 CVI_ISP_Init

【Description】

Initialize ISP firmware.

【Syntax】

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

【Parameter】

Parameter	Description
ViPipe	ViPipe number

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_Exit*

3.2.3 CVI_ISP_Run

【Description】

Run ISP firmware.

【Syntax】

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

【Parameter】

Parameter	Description
ViPipe	ViPipe number

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

3.2.4 CVI_ISP_RunOnce

【Description】

Run ISP firmware once.

【Syntax】

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

【Parameter】

Parameter	Description
ViPipe	ViPipe number

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

3.2.5 CVI_ISP_Exit

【Description】

Exit ISP firmware.

【Syntax】

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

【Parameter】

Parameter	Description
ViPipe	ViPipe number

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_Init*

3.2.6 CVI_ISP_SetPubAttr

【Description】

Set ISP public properties.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstPubAttr	ISP public properties.	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetPubAttr*

3.2.7 CVI_ISP_GetPubAttr

【Description】

Get ISP public properties.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstPubAttr	ISP public properties.	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetPubAttr*

3.2.8 CVI_ISP_SetFMWState

【Description】

Set ISP firmware status.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
enState	ISP firmware state	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetFMWState*

3.2.9 CVI_ISP_GetFMWState

【Description】

Get ISP firmware status.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
enState	ISP firmware state	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetFMWState*

3.2.10 CVI_ISP_SetModuleControl

【Description】

Set control of ISP function module.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
punModCtrl	Control of ISP function module	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetModuleControl*

3.2.11 CVI_ISP_GetModuleControl

【Description】

Get control of ISP function module.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
punModCtrl	Control of ISP function module	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetModuleControl*

3.2.12 CVI_ISP_GetVDTimeOut

【Description】

Get ISP interrupt information.

【Syntax】

```
CVI_S32 CVI_ISP_GetVDTimeOut(VI_PIPE ViPipe, ISP_VD_TYPE_E enIspVDType, CVI_U32␣
↪u32MilliSec);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
enIspVDType	Field synchronization signal	Input
u32MilliSec	Timeout time, in ms.	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

None.

3.2.13 CVI_ISP_SensorRegCallBack

【Description】

The callback interface of sensor registration provided by ISP.

【Syntax】

```
CVI_S32 CVI_ISP_SensorRegCallBack(VI_PIPE ViPipe, ISP_SNS_ATTR_INFO_S␣
↪*pstSnsAttrInfo, ISP_SENSOR_REGISTER_S *pstRegister);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstSnsAttrInfo	Properties of sensor registered with ISP	Input
pstRegister	Sensor register structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

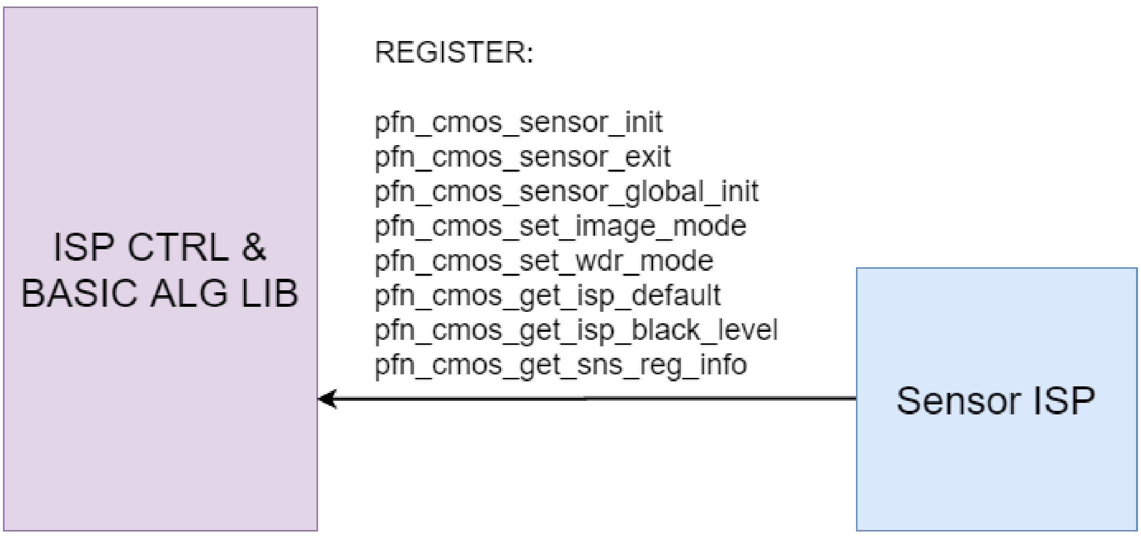


Fig. 3.1: Figure 2-1 Interface between ISP library and sensor Library

【Example】

None.

【Related Topic】

- *CVI_ISP_SensorUnRegCallback*

3.2.14 CVI_ISP_SensorUnRegCallBack

【Description】

The callback interface of sensor anti registration provided by ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
SensorId	The ID of the sensor registered with the ISP.	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_SensorRegCallBack*

3.2.15 CVI_ISP_AELibRegCallBack

【Description】

The callback interface of AE library registration provided by ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AWB library structure pointer	Input
pstRegister	AWB library register structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

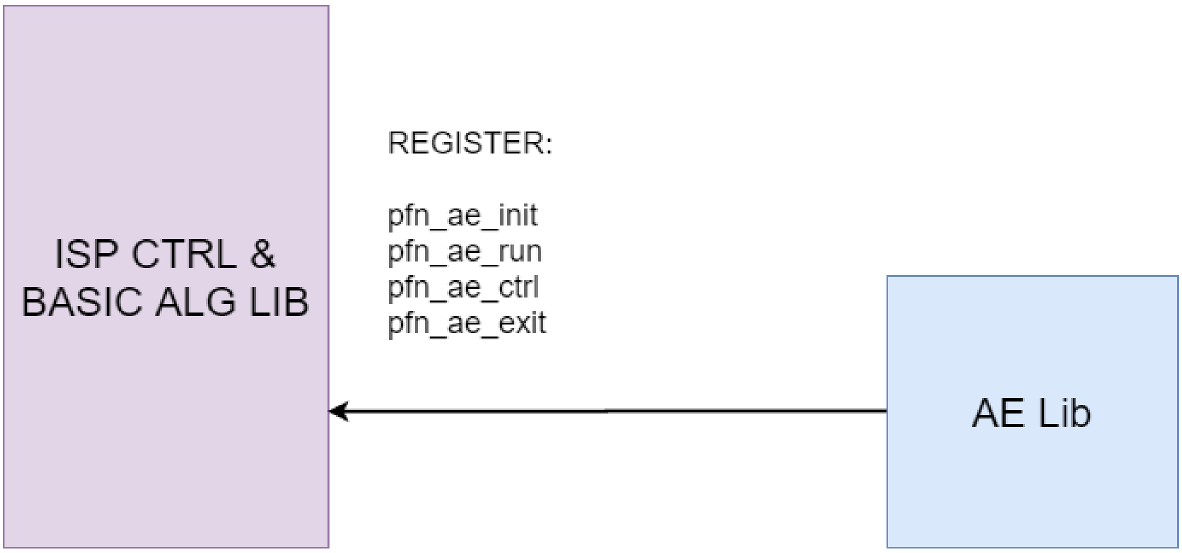


Fig. 3.2: Figure 2-2 Interface between ISP library and AE Library

【Example】

None.

【Related Topic】

- *CVI_ISP_AELibUnRegCallBack*

3.2.16 CVI_ISP_AELibUnRegCallBack

【Description】

The callback interface of AE library anti registration provided by ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AE library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_AELibRegCallBack*

3.2.17 CVI_ISP_AWBLibRegCallBack

【Description】

Callback interface for AWB library registration provided by ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAwbLib	AWB library structure pointer	Input
pstRegister	AWB library register structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

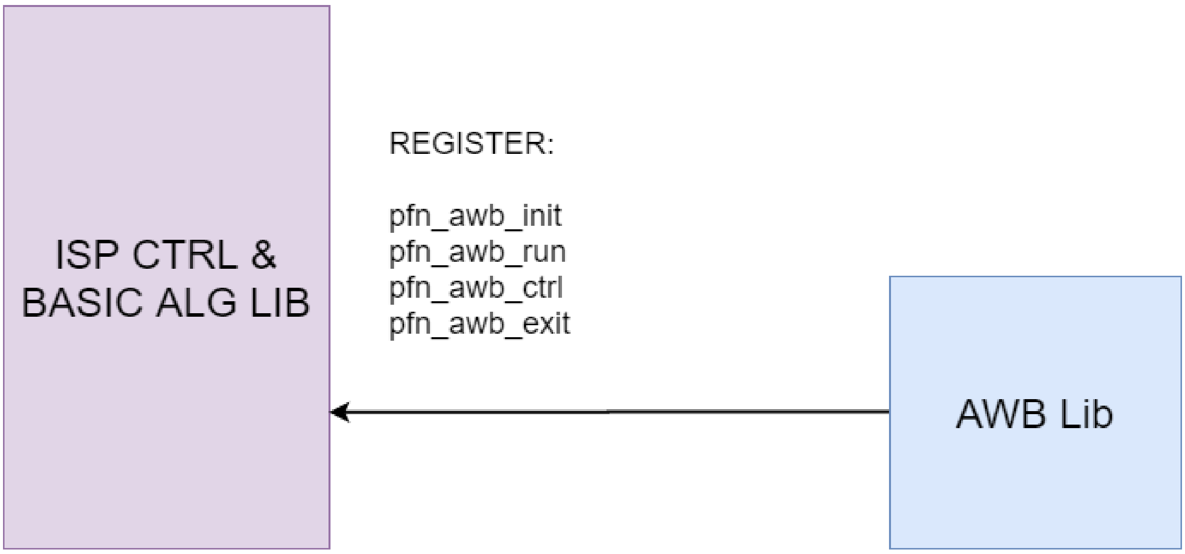


Fig. 3.3: Figure 2-3 Interface between ISP library and AWB Library

【Example】

None.

【Related Topic】

- *CVI_ISP_AWBLibUnRegCallBack*

3.2.18 CVI_ISP_AWBLibUnRegCallBack

【Description】

The callback interface of AWB library anti registration provided by ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAwbLib	AWB library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

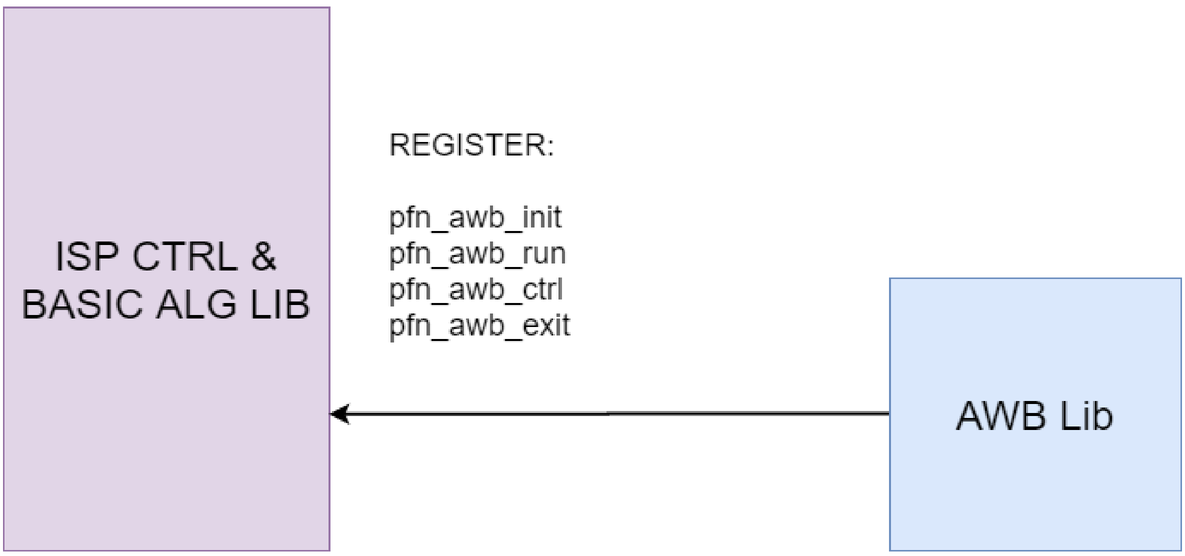


Fig. 3.4: Figure 2-3 Interface between ISP library and AWB Library

【Example】

None.

【Related Topic】

- *CVI_ISP_AWBLibRegCallBack*

3.2.19 CVI_ISP_SetBindAttr

【Description】

Set the binding relationship between ISP library, 3A library and sensor.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstBindAttr	Binding structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetBindAttr*

3.2.20 CVI_ISP_GetBindAttr

【Description】

Get the binding relationship between ISP library, 3A library and sensor.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstBindAttr	Binding structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

- This interface does not support multiprocess operations.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetBindAttr*

3.2.21 CVI_ISP_SetCtrlParam

【Description】

Set ISP control parameters.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIspCtrlParam	ISP control parameter structure pointer.	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCtrlParam*

3.2.22 CVI_ISP_GetCtrlParam

【Description】

Get ISP control parameters.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIspCtrlParam	ISP control parameter structure pointer.	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCtrlParam*

3.2.23 CVI_ISP_SetModParam

【Description】

Set ISP module parameters.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIspModParam	ISP module parameter structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetModParam*

3.2.24 CVI_ISP_GetModParam

【Description】

Get ISP module parameters.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
pstIspCtrlParam	ISP module parameter structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h
- Library files: libisp.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetModParam*

3.2.25 CVI_BIN_ExportBinData

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.26 CVI_BIN_ImportBinData

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.27 CVI_BIN_GetBinExtraAttr

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.28 CVI_BIN_GetBinTotalLen

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.29 CVI_BIN_SetBinName

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.30 CVI_BIN_GetBinName

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.31 CVI_BIN_SaveParamToBin

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.32 CVI_BIN_LoadParamFromBin

【Description】

For details, see section 3.1.2 of the 《PQ Tools User Guide》.

3.2.33 CVI_ISP_IrAutoRunOnce

【Description】

Run the infrared automatic switching function.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
IspDev	ISP device number	Input
pstIrAttr	Infrared automatic switching properties	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_comm_isp.h, cvi_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

```
ISP_DEV IspDev = 0;

ISP_IR_AUTO_ATTR_S stIrAttr;

stIrAttr.bEnable = 1;

stIrAttr.u32Normal2IrIsoThr = 3200;

stIrAttr.u32Ir2NormalIsoThr = 100;

stIrAttr.u32RGMin = 256;

stIrAttr.u32RGMax = 512;

stIrAttr.u32BGMin = 256;

stIrAttr.u32BGMax = 512;
```

(continues on next page)

(continued from previous page)

```
CVI_ISP_IrAutoRunOnce(IspDev, &stIrAttr);
```

【Related Topic】

None.

3.2.34 CVI_ISP_SetSmartInfo

【Description】

Set the coordinates recognized by Deep Learning (face, human form, object) for AE metering.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ISP device number	Input
pstSmartInfo	AE Deep Learning recognition coordinate information structure pointer	Input
TimeOut	The frame number of the recognition coordinate information is not updated, and the normal AE mode will be restored after exceeding	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
//Set the coordinate position of the face in the raw domain (X, Y , W, H) =(0,␣
↪0, 100, 100)
```

(continues on next page)

(continued from previous page)

```
// and frame width/height 1920/1080 information to AE

//Return to normal AE after the recognition result is not updated for 4 frames

VI_PIPE ViPipe = 0;

ISP_SMART_INFO_S stSmartInfo;

CVI_ISP_GetSmartInfo(ViPipe, &stSmartInfo);

stSmartInfo.stROI[0].bEnable = 1;

stSmartInfo.stROI[0].bAvailable = 1;

stSmartInfo.stROI[0].u8Num = 1;

stSmartInfo.stROI[0].u16PosX[0] = 0;

stSmartInfo.stROI[0].u16PosY[0] = 0;

stSmartInfo.stROI[0].u16Width[0] = 100;

stSmartInfo.stROI[0].u16Height[0] = 100;

stSmartInfo.stROI[0].u16FrameWidth = 1920;

stSmartInfo.stROI[0].u16FrameHeight = 1080;

CVI_ISP_SetSmartInfo(ViPipe, &stSmartInfo, 4);
```

【Related Topic】

- *CVI_ISP_GetSmartInfo*

3.2.35 CVI_ISP_GetSmartInfo

【Description】

Get the coordinates recognized by Deep Learning (face, human form, object) for AE metering.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ISP device number	Input
pstSmartInfo	AE Deep Learning recognition coordinate information structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetSmartInfo*

3.3 Data Types

Variables in this document that do not explicitly specify a range of values default to the range of values corresponding to the data type. For example, Variables of CVI_U8 data type range from [0, 255]. Variables in this document, such as data precision, are not explicitly specified. Default is 1.

3.3.1 RECT_S

【Description】

Define the start position of the clipping window and the image width and height

【Syntax】

```
typedef struct _RECT_S {

CVI_S32 s32X;

CVI_S32 s32Y;

CVI_U32 u32Width;

CVI_U32 u32Height;

} RECT_S;
```

【Member】

Member	Description
s32X	Horizontal starting position
s32Y	Vertical starting position
u32Width	Image Width
u32Height	Image Height

【Note】

None.

【Related Data Type and Interface】

None.

3.3.2 SIZE_S

【descriptiob】

Defines the width and height of the sensor output.

【Syntax】

```
typedef struct _SIZE_S {

CVI_U32 u32Width;

CVI_U32 u32Height;

} SIZE_S;
```

【Member】

Member	Description
u32Width	Sensor output width
u32Height	Sensor output height

【Note】

None.

【Related Data Type and Interface】

None.

3.3.3 ISP_BAYER_FORMAT_E

【Description】

Defines the format type of the Bayer array for the input ISP image

【Syntax】

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

【Member】

Member	Description
BAYER_XX	The format types of the various Bayer arrays, and the names indicate how the pixels are arranged

【Note】

None.

【Related Data Type and Interface】

None.

3.3.4 WDR_MODE_E

【Description】

Defines the operation mode of the sensor

【Syntax】

```
typedef enum _WDR_MODE_E {
  WDR_MODE_NONE = 0,
  WDR_MODE_BUILT_IN,
  WDR_MODE_QUDRA,
```

(continues on next page)

(continued from previous page)

```
  WDR_MODE_2To1_LINE,
  WDR_MODE_2To1_FRAME,
  WDR_MODE_2To1_FRAME_FULL_RATE,

  WDR_MODE_3To1_LINE,
  WDR_MODE_3To1_FRAME,
  WDR_MODE_3To1_FRAME_FULL_RATE,
  WDR_MODE_4To1_LINE,
  WDR_MODE_4To1_FRAME,
  WDR_MODE_4To1_FRAME_FULL_RATE,

  WDR_MODE_MAX,
} WDR_MODE_E;
```

【Member】

Member	Description
WDR_MODE_NONE	Linear mode.
WDR_MODE_BUILT_IN	Sensor synthetic WDR mode.
WDR_MODE_QUDRA	Qudra mode
WDR_MODE_2To1_LINE	2-frame composite row WDR mode.
WDR_MODE_2To1_FRAME	2-frame composite frame WDR mode.
WDR_MODE_2To1_FRAME_FULL_RATE	2-frame Composite Frame WDR Full Frame Rate Mode.
WDR_MODE_3To1_LINE	3-frame composite row WDR mode
WDR_MODE_3To1_FRAME	3-frame composite frame WDR mode.
WDR_MODE_3To1_FRAME_FULL_RATE	3-frame Composite Frame WDR Full Frame Rate Mode.
WDR_MODE_4To1_LINE	4-frame composite row WDR mode.
WDR_MODE_4To1_FRAME	4-frame composite frame WDR mode.
WDR_MODE_4To1_FRAME_FULL_RATE	4-frame Composite Frame WDR Full Frame Rate Mode.

【Note】

None.

【Related Data Type and Interface】

None.

3.3.5 ISP_PUB_ATTR_S

【Description】

Define ISP Common Properties

【Syntax】

```
typedef struct _ISP_PUB_ATTR_S {
  RECT_S stWndRect;
```

(continues on next page)

(continued from previous page)

```
  SIZE_S stSnsSize;
  CVI_FLOAT f32FrameRate;
  ISP_BAYER_FORMAT_E enBayer;
  WDR_MODE_E enWDRMode;
  CVI_U8 u8SnsMode;
} ISP_PUB_ATTR_S;
```

【Member】

Member	Description
stWndRect	rect of Isp output
stSnsSize	image size of sensor
f32FrameRate	Sensor frame rate
enBayer	Bayer format for Sensor
enWDRMode	Wide dynamic mode
u8SnsMode	Used for selection of Sensor initialization sequences, different u8SnsModes are configured to correspond to different initialization sequences at the same resolution and frame rate.Otherwise, u8SnsModel is configured as 0 by default and can be selected through stSnsSize and f32FrameRate for initialization sequence selection.

【Note】

None.

【Related Data Type and Interface】

- *RECT_S*
- *SIZE_S*
- *ISP_BAYER_FORMAT_E*
- *WDR_MODE_E*
- *CVI_ISP_SetPubAttr*
- *CVI_ISP_GetPubAttr*

3.3.6 ISP_FMW_STATE_E

【Description】

Define the ISP firmware state.

【Syntax】

```
typedef enum _ISP_FMW_STATE_E {
  ISP_FMW_STATE_RUN,
  ISP_FMW_STATE_FREEZE,
```

(continues on next page)

(continued from previous page)

```
  ISP_FMW_STATE_BUTT
} ISP_FMW_STATE_E;
```

【Member】

Member	Description
ISP_FMW_STATE_RUN	The Firmware runs properly
ISP_FMW_STATE_FREEZE	Firmware Freeze State

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetFMWState*
- *CVI_ISP_GetFMWState*

3.3.7 ISP_MODULE_CTRL_U

【Description】

Define control of ISP function modules

【Syntax】

```
typedef union _ISP_MODULE_CTRL_U {
  CVI_U64 u64Key;
  struct {
    CVI_U64 bitBypassBlc : 1; /* RW;[0] */
    CVI_U64 bitBypassRlsc : 1; /* RW;[1] */
    CVI_U64 bitBypassFpn : 1; /* RW;[2] */
    CVI_U64 bitBypassDpc : 1; /* RW;[3] */
    CVI_U64 bitBypassCrosstalk : 1; /* RW;[4] */
    CVI_U64 bitBypassWBGain : 1; /* RW;[5] */
    CVI_U64 bitBypassDis : 1; /* RW;[6] */
    CVI_U64 bitBypassBnr : 1; /* RW;[7] */
    CVI_U64 bitBypassDemosaic : 1; /* RW;[8] */
    CVI_U64 bitBypassRbgcac : 1; /* RW;[9] */
    CVI_U64 bitBypassMlsc : 1; /* RW;[10] */
    CVI_U64 bitBypassCcm : 1; /* RW;[11] */
    CVI_U64 bitBypassFusion : 1; /* RW;[12] */
    CVI_U64 bitBypassDrc : 1; /* RW;[13] */
    CVI_U64 bitBypassGamma : 1; /* RW;[14] */
    CVI_U64 bitBypassDehaze : 1; /* RW;[15] */
    CVI_U64 bitBypassClut : 1; /* RW;[16] */
    CVI_U64 bitBypassCsc : 1; /* RW;[17] */
    CVI_U64 bitBypassDci : 1; /* RW;[18] */
    CVI_U64 bitBypassCa : 1; /* RW;[19] */
```

(continues on next page)

(continued from previous page)

```
    CVI_U64 bitBypassPreyee : 1; /* RW;[20] */
    CVI_U64 bitBypassMotion : 1; /* RW;[21] */
    CVI_U64 bitBypass3dnr : 1; /* RW;[22] */
    CVI_U64 bitBypassYnr : 1; /* RW;[23] */
    CVI_U64 bitBypassCnr : 1; /* RW;[24] */
    CVI_U64 bitBypassCac : 1;   /*RW:[25]*/
    CVI_U64 bitBypassCa2 : 1;   /*RW:[26]*/
    CVI_U64 bitBypassYee : 1;   /*RW:[27]*/
    CVI_U64 bitBypassYcontrast : 1;   /*RW:[28]*/
    CVI_U64 bitBypassMono : 1;   /*RW:[29]*/
    CVI_U64 bitRsv : 34; /* H  ; [30:63] */
  };
} ISP_MODULE_CTRL_U;
```

【Member】

Member	Description
u64Key	Integer value for structure enumeration
bitbypassxxx	Each module function controls bit

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetModuleControl*
- *CVI_ISP_GetModuleControl*

3.3.8 ISP_VD_TYPE_E

【Description】

Define synchronization signal with ISP

【Syntax】

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

【Member】

Member	Description
ISP_VD_FE_START	Starting signal of FE frame
ISP_VD_FE_END	End signal of FE frame
ISP_VD_BE_END	End signal of BE frame

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_GetVDTimeOut*

3.3.9 ISP_SNS_ATTR_INFO_S

【Description】

Define ISP sensor properties.

【Syntax】

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

【Member】

Member	Description
eSensorid	Sensor ID number

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SensorRegCallBack*

3.3.10 ALG_LIB_S

【Description】

Used library information

【Syntax】

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

【Member】

Member	Description
s32Id	Id of the algorithm library instance.
acLibName	An array of characters identifying the name of the algorithm library.

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SensorRegCallBack*
- *CVI_ISP_SensorUnRegCallBack*
- *CVI_ISP_AELibRegCallBack*
- *CVI_ISP_AELibUnRegCallBack*
- *CVI_ISP_AWBLibRegCallBack*
- *CVI_ISP_AWBLibUnRegCallBack*

3.3.11 ISP_AE_EXP_FUNC_S

【Description】

Defines the AE callback function structure.

【Syntax】

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

【Member】

Member	Description
pfn_ae_init	Initializes the callback function pointer for AE.
pfn_ae_run	Callback function pointer that runs AE.
pfn_ae_ctrl	Callback function pointer that controls the internal state of AE.
pfn_ae_exit	Destroy the callback function pointer for AE.

【Note】

None.

【Related Data Type and Interface】

- *ISP_AE_REGISTER_S*

3.3.12 ISP_AE_REGISTER_S

【Description】

【Syntax】

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

【Member】

Member	Description
stAeExpFunc	AE registered callback function structure.

【Note】

None.

【Related Data Type and Interface】

- *ISP_AE_EXP_FUNC_S*
- *CVI_ISP_AELibRegCallBack*

3.3.13 ISP_SENSOR_EXP_FUNC_S

【Description】

Define sensor callback function structure

【Syntax】

```
typedef struct _ISP_SENSOR_EXP_FUNC_S {
  CVI_VOID (*pfn_cmos_sensor_init)(VI_PIPE ViPipe);
  CVI_VOID (*pfn_cmos_sensor_exit)(VI_PIPE ViPipe);
  CVI_VOID (*pfn_cmos_sensor_global_init)(VI_PIPE ViPipe);
  CVI_S32 (*pfn_cmos_set_image_mode)(VI_PIPE ViPipe, ISP_CMOS_SENSOR_IMAGE_MODE_
↪S *pstSensorImageMode);
  CVI_S32 (*pfn_cmos_set_wdr_mode)(VI_PIPE ViPipe, CVI_U8 u8Mode);

  /* the algs get data which is associated with sensor, except 3a */
  CVI_S32 (*pfn_cmos_get_isp_default)(VI_PIPE ViPipe, ISP_CMOS_DEFAULT_S␣
↪*pstDef);
  CVI_S32 (*pfn_cmos_get_isp_black_level)(VI_PIPE ViPipe, ISP_CMOS_BLACK_LEVEL_
↪S *pstBlackLevel);
  CVI_S32 (*pfn_cmos_get_sns_reg_info)(VI_PIPE ViPipe, ISP_SNS_SYNC_INFO_S␣
↪*pstSnsRegsInfo);

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

【Member】

Member	Description
pfn_cmos_sensor_init	Initialize sensor' s callback function pointer.
pfn_cmos_sensor_exit	Sensor' s callback exits the function pointer.
pfn_cmos_sensor_global_init	Initializes a callback function pointer to a global variable.
pfn_cmos_set_image_mode	Sets the callback function pointer for resolution and frame rate switching.
pfn_cmos_set_wdr_mode	Sets the callback function pointer in wdr mode.
pfn_cmos_get_isp_default	A callback function pointer that gets the initial value of an ISP-based algorithm.
pfn_cmos_get_isp_black_level	A callback function pointer to get the black level value of the sensor, which supports dynamic adjustment of the black level value based on the sensor gain.
pfn_cmos_get_sns_reg_info	A callback function pointer for sensor register information to implement kernel state configuration AE information.
pfn_cmos_set_pixel_detect	Sets the callback function pointer of the bad point correction switch.

【Note】

None.

【Related Data Type and Interface】

None.

3.3.14 ISP_SENSOR_REGISTER_S

【Description】

【Syntax】

```
typedef struct bmISP_SENSOR_REGISTER_S {

  ISP_SENSOR_EXP_FUNC_S stSnsExp;

} ISP_SENSOR_REGISTER_S;
```

【Member】

Member	Description
stSnsExp	Sensor registered callback function structure.

【Note】

None.

【Related Data Type and Interface】

- *ISP_SENSOR_EXP_FUNC_S*

- *CVI_ISP_SensorRegCallBack*

3.3.15 ISP_AWB_EXP_FUNC_S

【Description】

【Syntax】

```
typedef struct _ISP_AWB_EXP_FUNC_S {

  CVI_S32 (*pfn_awb_init)(VI_PIPE ViPipe, const ISP_AWB_PARAM_S *pstAwbParam);

  CVI_S32 (*pfn_awb_run)(VI_PIPE ViPipe, const ISP_AWB_INFO_S *pstAwbInfo, ISP_
↪AWB_RESULT_S *pstAwbResult,

  CVI_S32 s32Rsv);

  CVI_S32 (*pfn_awb_ctrl)(VI_PIPE ViPipe, CVI_U32 u32Cmd, CVI_VOID * pValue);

  CVI_S32 (*pfn_awb_exit)(VI_PIPE ViPipe);

} ISP_AWB_EXP_FUNC_S;
```

【Member】

【Note】

None.

【Related Data Type and Interface】

None.

3.3.16 ISP_AWB_REGISTER_S

【Description】

【Syntax】

```
typedef struct _ISP_AWB_REGISTER_S {

  ISP_AWB_EXP_FUNC_S stAwbExpFunc;

} ISP_AWB_REGISTER_S;
```

【Member】

【Note】

None.

【Related Data Type and Interface】

- *ISP_AWB_EXP_FUNC_S*
- *CVI_ISP_AWBLibRegCallBack*

3.3.17 ISP_BIND_ATTR_S

【Description】

【Syntax】

```
typedef struct _ISP_BIND_ATTR_S {

  CVI_S32 sensorId;

  ALG_LIB_S stAeLib;

  ALG_LIB_S stAfLib;

  ALG_LIB_S stAwbLib;

} ISP_BIND_ATTR_S;
```

【Member】

Member	Description
SensorId	registered Sensor Id
stAeLib	AE library structure.
stAwbLib	AF library structure.
stAfLib	AWB library structure.

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetBindAttr*
- *CVI_ISP_GetBindAttr*

3.3.18 ISP_CTRL_PARAM_S

【Description】

Define the ISP control parameter structure.

【Syntax】

```
typedef struct _ISP_CTRL_PARAM_S {

  CVI_U32 u32AEStatIntvl;
```

(continues on next page)

(continued from previous page)

```
    CVI_U32 u32AWBStatIntvl;
    CVI_U32 u32AFStatIntvl;
    CVI_U32 u32ProcParam;
    CVI_U32 u32ProcLevel;
    CVI_U32 u32UpdatePos;
    CVI_U32 u32IntTimeOut;
    CVI_U32 u32PwmNumber;
    CVI_U32 u32PortIntDelay;
} ISP_CTRL_PARAM_S;
```

【Member】

Member	Description
u32AEStatIntvl	ISP 3A AE statistics update frequency, unit is frame. Value range: (0,0xffffffff]
u32AWBStatIntvl	Update frequency of ISP 3A AWB statistics in frames. Range of values: (0,0xffffff]
u32AFStatIntvl	Update frequency of ISP 3A AF statistics in frames. Range of values: (0,0xffffff]
u32ProcParam	The frequency at which ISP collects information, in frames, with a default value of 30. The higher this value, the lower the RISC-V utilization of the ISP. Range of values: (0,0xffffff]
u32ProcLevel	ISP’s proc prints Level, Level 0, proc function is off, Level 1, Simplified information, Level 2, More information, Level 3, Very much information (will also print 3A statistics)
u32UpdatePos	Currently cvitek only supports configuring sensor registers at the beginning of a frame with a default value of 0
u32IntTimeOut	Represents the time (ms) at which the interrupt timed out. Cvitek is not currently in use
u32PwmNumber	Represents a PWM number. Cvitek is not currently in use
u32PortIntDelay	Represents Port interrupt delay time

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCtrlParam*
- *CVI_ISP_GetCtrlParam*

3.3.19 ISP_MOD_PARAM_S

【Description】

Define ISP module parameter structure.

【Syntax】

```
typedef struct _ISP_MOD_PARAM_S {

 CVI_U32 u32IntBotHalf;

} ISP_MOD_PARAM_S;
```

【Member】

Member	Description
u32intbothalf	Indicates whether the ISP interrupt processing adopts the lower half mechanism, and the default value is 0. Currently, cvitek only supports (reading statistics and configuring sensor and ISP synchronization registers) to be completed in the interrupt service routine;

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetModParam*
- *CVI_ISP_GetModParam*

3.3.20 ISP_IR_AUTO_ATTR_S

【Description】

Defines the infrared auto-switching properties.

【Syntax】

```
typedef struct _ISP_IR_AUTO_ATTR_S

{

 CVI_BOOL bEnable;

 CVI_U32 u32Normal2IrIsoThr;

 CVI_U32 u32Ir2NormalIsoThr;
```

(continues on next page)

(continued from previous page)

```
    CVI_U32 u32RGMax;

    CVI_U32 u32RGMin;

    CVI_U32 u32BGMax;

    CVI_U32 u32BGMin;

    ISP_IR_STATUS_E enIrStatus;

    ISP_IR_SWITCH_STATUS_E enIrSwitch;

} ISP_IR_AUTO_ATTR_S;
```

【Member】

Member	Description
benable:	Infrared automatic switching enable. CVI_FALSE: off; CVI_TRUE: enable.
u32Normal2IrIsoThr	The ISO threshold for switching from normal to infrared. When the actual effective ISO is greater than this threshold, the system needs to switch to the infrared state. Value range: [0, 0xFFFFFFFF].
u32Ir2NormalIsoThr	The ISO threshold for switching from infrared state to normal state. When the actual effective ISO is lower than this threshold, the system needs to switch to the normal state. Value range: [0, 0xFFFFFFFF].
u32RGMax	R/G maximum value in infrared state. When the R/G of the actual image is greater than this parameter, the system needs to switch to the normal state. 4.8 format. Value range: [0, 0xFFF].
u32RGMin	R/G minimum value in infrared state. When the R/G of the actual image is smaller than this parameter, the system needs to switch to the normal state. 4.8 format. Value range: [0, u32RGMax].
u32BGMax	The maximum value of B/G in the infrared state. When the B/G of the actual image is greater than this parameter, the system needs to switch to the normal state. 4.8 format. Value range: [0, 0xFFF].
u32BGMin	B/G minimum value in infrared state. When the B/G of the actual image is smaller than this parameter, the system needs to switch to the normal state. 4.8 format. Value range: [0, u32BGMax].
enIrStatus	The current infrared state of the device. It should be configured as the actual infrared state of the device, and the user needs to ensure the correctness of the state.
enIrSwitch	The infrared switching status of the device, which is read-only.

【Note】

None.

【Related Data Type and Interface】

None.

4 AE

4.1 Overview

The function of the CVI AE module is to get the exposure of the current image according to the auto-metering system, and then automatically configure the lens aperture, sensor shutter and gain to obtain the best image quality.The main algorithms for automatic exposure are aperture priority, shutter priority and gain priority.

When the aperture is preferred, the algorithm will first adjust the aperture to the appropriate position, and then allocate the exposure time and gain to only fit the p-iris lens, which will balance the noise and depth of field.When shutter priority occurs, the algorithm assigns exposure time first, then sensor gain and ISP gain, so that the image noise taken is less.Gain priority is to assign sensor gain and ISP gain first, and then to assign exposure time, which is suitable for shooting scenes of moving objects.The current AE algorithm also supports clients to set more flexible exposure allocation strategies, and the workflow of the AE module is shown in Fig. 4.1 .

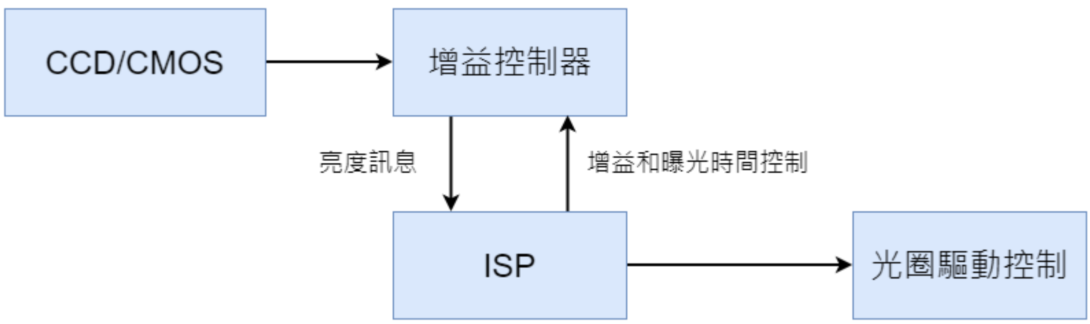


Fig. 4.1: AE module workflow diagram

4.2 Important Concept

- Exposure time: The time when the sensor accumulates charge is the period from the beginning of the exposure of the sensor pixel to the time when the charge is read out.
- Exposure gain: The total amplification factor of the sensor’s output charge generally has digital gain and analog gain. The noise introduced by analog gain is slightly smaller, so analog gain is generally preferred.

- Aperture: Aperture is a mechanical device in the lens that can change the size of the middle hole.
- Anti-flicker: Picture flicker caused by the mismatch between the power frequency of the lamp and the frame rate of the sensor. Anti-flicker effect is usually achieved by limiting the exposure time and modifying the frame rate of the sensor.

4.3 Function Overview

The AE module mainly consists of the AE statistical information module of ISP and the AE algorithm Firmware of AE control strategy. The AE statistical information module of ISP mainly provides brightness information statistics of sensor input data. The statistical information provided includes histogram and mean value. It also provides histogram and R/Gr/Gb/B four-component mean of 256 segments of the whole image. It also provides four-component mean R/Gr/Gb/B statistics of dividing the whole image into MxN blocks, each block, as shown in Fig. 4.2 .

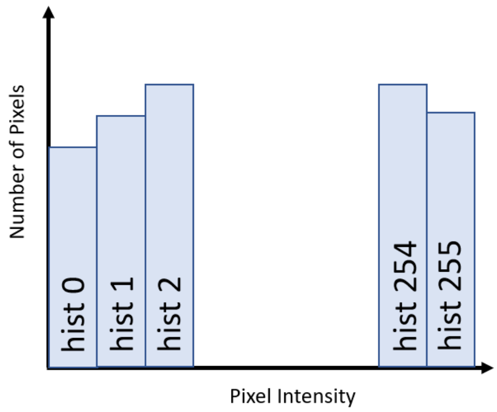


Fig. 4.2: AE 256 statistical information histogram

The main working principle of AE algorithm is to obtain the statistical information of the input image and compare it with the set target brightness instantly, while dynamically adjusting the exposure time and gain of the sensor and the lens aperture size to achieve the actual brightness close to the set target brightness. Its working principle is shown in Fig. 4.3 .

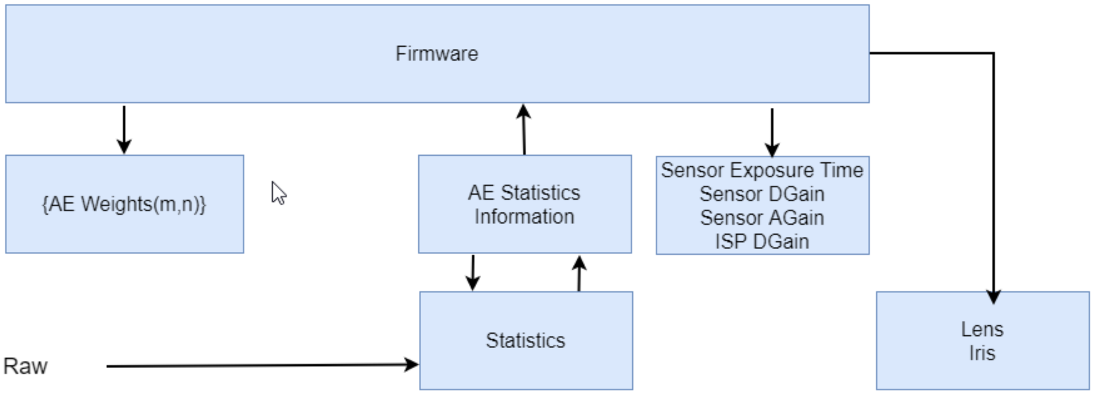


Fig. 4.3: AE working principle

4.4 API Reference

4.4.1 AE Library Interface

All AE library interfaces are for CVI AE libraries only. If the client implements the AE libraries themselves, they do not need to be concerned about these interfaces, and they cannot be used.

- *CVI_AE_Register* :Register AE library with ISP.
- *CVI_AE_UnRegister* :Reverse registration of AE libraries with ISP.
- *CVI_AE_SensorRegCallBack* : Callback interface for sensor registration provided by AE Library
- *CVI_AE_SensorUnRegCallBack* : The sensor deregistered callback interface provided by the AE library.

4.4.1.1 CVI_AE_Register

【Description】

Register AE library with ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AE algorithm library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_AE_UnRegister*

4.4.1.2 CVI_AE_UnRegister

【Description】

Reverse registration of AE libraries with ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AE algorithm library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- This interface invokes the AE anti-registration callback interface provided by the ISP library to implement the function of AE anti-registration with the ISP library.
- This interface does not support multiprocess operations.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_AE_Register*

4.4.1.3 CVI_AE_SensorRegCallBack

【Description】

The sensor registered callback interface provided by the AE library.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AE algorithm library structure pointer	Input
pstSnsAttrInfo	Properties of the Sensor registered with AE	Input
pstRegister	Sensor Registration Structure Pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- SensorId is a custom value in the sensor library that is used to proofread whether the sensor registered with ISP and the sensor registered with 3A are the same sensor.
- AE obtains the differential initialization parameters and controls the sensor through a series of callback interfaces registered with the sensor.
- This interface does not support multiprocess operations.

- This interface is not supported on the linux side of the dual-os SDK.

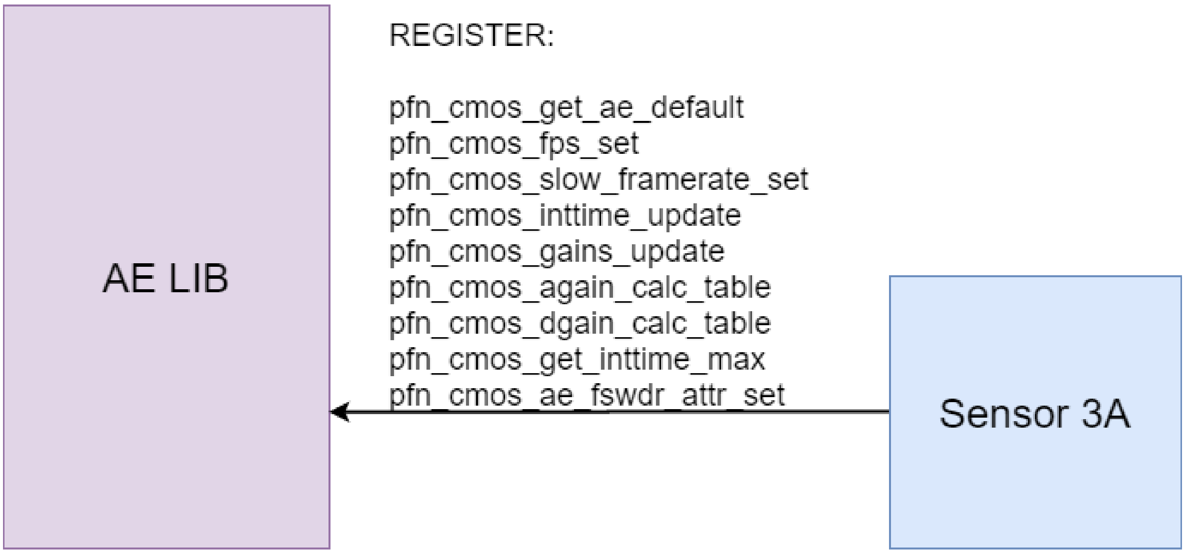


Fig. 4.4: Interface between AE library and sensor Library

【Example】

None.

【Related Topic】

- *CVI_AE_SensorUnRegCallBack*

4.4.1.4 CVI_AE_SensorUnRegCallBack

【Description】

The sensor deregistered callback interface provided by the AE library.

【Syntax】

```
CVI_S32 CVI_AE_SensorUnRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAeLib, SENSOR_
↪ID SensorId);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AE algorithm library structure pointer	Input
SensorId	The Id of the Sensor that is unregistered with AE	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae_comm.h
- Library files: libae.a

【Note】

- SensorId is a custom value in the sensor library that is used to proofread whether the sensor registered with ISP and the sensor registered with 3A are the same sensor.
- This interface does not support multiprocess operations.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None.

【Related Topic】

- *CVI_AE_SensorRegCallBack*

4.4.2 AE control module

Exposure Control Interface:

- *CVI_ISP_SetExposureAttr* : Set AE exposure properties.
- *CVI_ISP_GetExposureAttr* : Get the AE exposure properties.
- *CVI_ISP_SetWDRExposureAttr* : Set the AE exposure properties in WDR mode.
- *CVI_ISP_GetWDRExposureAttr* : Get the AE exposure properties in WDR mode.
- *CVI_ISP_SetAERouteAttr* : Set AE route properties.
- *CVI_ISP_GetAERouteAttr* : Get AE route properties.
- *CVI_ISP_SetAERouteAttrEx* : Set the extended properties of AE exposure allocation and support setting the sensor analog gain, sensor digital gain and ISP digital gain in AE allocation policy respectively.
- *CVI_ISP_GetAERouteAttrEx* : Getting AE Exposure Allocation Policy Extended Properties,
- *CVI_ISP_QueryExposureInfo* : Get AE internal status information.
- *CVI_ISP_SetAntiFlicker* : Set AE anti flicker function.
- *CVI_ISP_GetAntiFlicker* : Get AE anti flicker settings.
- *CVI_ISP_QueryFps* : Get the current fps of AE.
- *CVI_ISP_GetCurrentLvX100* : Get the LV value of the current ambient brightness.

- *CVI_ISP_SetSmartExposureAttr* : Set the AE exposure attribute in smart mode.
- *CVI_ISP_GetSmartExposureAttr* : Get the AE exposure attribute in smart mode.
- *CVI_ISP_SetAERouteSFAttr* : Set AE WDR short detection exposure allocation strategy.
- *CVI_ISP_GetAERouteSFAttr* : Get AE WDR short detection exposure allocation strategy.
- *CVI_ISP_SetAERouteSFAttrEx* : Set AE WDR short detection exposure allocation extended attributes.
- *CVI_ISP_GetAERouteSFAttrEx* : Get AE WDR short detection exposure allocation extended attributes.

4.4.2.1 CVI_ISP_SetExposureAttr

【Description】

Set AE exposure properties.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstExpAttr	AE Exposure Attribute Structure Pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- When the AE exposure control type is automatic, the exposure time and the exposure gain are automatically controlled by the AE algorithm. Different exposure effects can be obtained by configuring the parameters inside the auto-exposure attribute structure stAuto.
- When the AE exposure control type is manual, it is possible to configure the manual exposure attribute structure stManual control to enable the type (Exposure time enable, ISO num enable, sensor analog gain enable, sensor digital gain enable, ISP digital gain enable) and corresponding exposure parameters(Exposure time, ISO Num, Sensor analog gain, Sensor digital gain, ISP digital gain).

- When the AE exposure control type is automatic, the parameters for configuring the manual exposure property are invalid.Similarly, when the AE exposure control type is manual, the parameters for configuring the auto-exposure properties are invalid.
- When the AE exposure control type is manual, if the exposure parameter is set beyond the maximum (small) value, the maximum (small) value supported by sensor will be used instead.
- Whether it is automatic or manual exposure, the exposure time is in microseconds (us), and the exposure gain is in multiples of 10 bit precision, i.e., 1024 is 1 times, 2048 is 2 times, etc.

【Example】

```
// Automatic exposure property setting
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Max = 40000;
stExpAttr.stAuto.stExpTimeRange.u32Min = 10;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

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

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

stExpAttr.stAuto.stAntiflicker.bEnable = CVI_TRUE;
stExpAttr.stAuto.stAntiflicker.enFrequency = AE_FREQUENCE_50HZ;
stExpAttr.stAuto.stAntiflicker.enMode = ISP_ANTIFLICKER_NORMAL_MODE;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

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

//Manual exposure property setting (use gain control):
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enAGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enDGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISPDGainOpType = OP_TYPE_MANUAL;
```

(continues on next page)

(continued from previous page)

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

//Manual exposure property setting (use ISO Num control):
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISONumOpType= OP_TYPE_MANUAL;

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

// Automatic mode sets the maximum gain to 32x (use gain control)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.enGainType = AE_TYPE_GAIN;
stExpAttr.stAuto.stSysGainRange.u32Max = 32767;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

// Automatic mode set maximum gain 32x (using ISO Num control)VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.enGainType = AE_TYPE_ISO;
stExpAttr.stAuto.stISONumRange.u32Max = 3200;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//Shutter mode (33333 us) set the maximum gain 32xVI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Min = 33333;
```

(continues on next page)

(continued from previous page)

```
stExpAttr.stAuto.stExpTimeRange.u32Max = 33333;
stExpAttr.stAuto.enGainType = AE_TYPE_ISO;
stExpAttr.stAuto.stISONumRange.u32Max = 3200;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//Set fixed gain in manual mode (32x, use ISO Num)VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enGainType = AE_TYPE_ISO;
stExpAttr.stManual.enISONumOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.u32ISONum = 3200;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//Set fixed gain in manual mode (32x, use Gain)VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enGainType = AE_TYPE_ISO;
stExpAttr.stManual.enAGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enDGainOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.enISPDGainOpType = OP_TYPE_MANUAL;

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

//automatic shutter
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

// Manually set the shutter (16384 us)
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.stManual.enExpTimeOpType = OP_TYPE_MANUAL;
stExpAttr.stManual.u32ExpTime = 16384;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//shutter mode (16384 us)
```

(continues on next page)

(continued from previous page)

```
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.bByPass = CVI_FALSE;
stExpAttr.enOpType = OP_TYPE_AUTO;
stExpAttr.stAuto.stExpTimeRange.u32Min = 16384;
stExpAttr.stAuto.stExpTimeRange.u32Max = 16384;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//Set antiflicker 50Hz
VI_PIPE ViPipe = 0;
ISP_EXPOSURE_ATTR_S stExpAttr;
CVI_ISP_GetExposureAttr(ViPipe, &stExpAttr);
stExpAttr.stAuto.stAntiflicker.bEnable = 1;
stExpAttr.stAuto.stAntiflicker.enMode = ISP_ANTIFLICKER_NORMAL_MODE;
stExpAttr.stAuto.stAntiflicker.enFrequency = AE_FREQUENCE_50HZ;
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);

//Set the target brightness of each LV
#define TABLE_SIZE  21
CVI_U8 target_max[TABLE_SIZE] = {15, 15, 15, 15,
  15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60, 60,
  60, 60};
CVI_U8 target_min[TABLE_SIZE] = {40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45,
      50, 50, 50, 50, 50, 50, 60, 60, 60};

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

for (i = 0; i < TABLE_SIZE;i++) {
  stExpAttr.au8AdjustTargetMin[i] =
  stExpAttr.au8AdjustTargetMax[i] = target [i];

}
CVI_ISP_SetExposureAttr(ViPipe, &stExpAttr);
```

【Related Topic】

- *CVI_ISP_GetExposureAttr*

4.4.2.2 CVI_ISP_GetExposureAttr

【Description】

Gets the AE exposure properties.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstExpAttr	AE Exposure Attribute Structure Pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetExposureAttr*

4.4.2.3 CVI_ISP_SetWDRExposureAttr

【Description】

Set the AE exposure properties in WDR mode.

【Syntax】

```
CVI_S32 CVI_ISP_SetWDRExposureAttr(VI_PIPE ViPipe, const ISP_WDR_EXPOSURE_ATTR_
↪S *pstWDRExpAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstWDRExpAttr	AE Exposure Attribute Structure Pointer in WDR Mode	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
//Manually set the exposure ratio to 8x
VI_PIPE ViPipe = 0;

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

//Set the auto mode exposure ratio to be limited to 8x ~ 16x
ISP_WDR_EXPOSURE_ATTR_S stWdrExpAttr;
CVI_ISP_GetWDRExposureAttr(ViPipe, &stWdrExpAttr);
stWdrExpAttr.enExpRatioType = OP_TYPE_AUTO;
stWdrExpAttr.u32ExpRatioMin = 512;
stWdrExpAttr.u32ExpRatioMax = 1024;
CVI_ISP_SetWDRExposureAttr(ViPipe, &stWdrExpAttr);

//Set the target brightness of each LV in the long/short frame
#define TABLE_SIZE  21
CVI_U8 LeTarget[TABLE_SIZE] = {15, 15, 15, 15,
  15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60, 60,
  60, 60};

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

(continues on next page)

(continued from previous page)

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

for (i = 0; i < TABLE_SIZE;i++) {
  stWdrExpAttr.au8LEAdjustTargetMin[i] =
  stWdrExpAttr.au8LEAdjustTargetMax[i] = LeTarget[i];

  stWdrExpAttr.au8SEAdjustTargetMin[i] =
  stWdrExpAttr.au8SEAdjustTargetMax[i] = SeTarget[i];

}
CVI_ISP_SetWDRExposureAttr(ViPipe, &stWdrExpAttr);
```

【Related Topic】

- *CVI_ISP_GetWDRExposureAttr*

4.4.2.4 CVI_ISP_GetWDRExposureAttr

【Description】

Gets the AE exposure properties in WDR mode.

【Syntax】

```
CVI_S32 CVI_ISP_GetWDRExposureAttr(VI_PIPE ViPipe, ISP_WDR_EXPOSURE_ATTR_S␣
↪*pstWDRExpAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstWDRExpAttr	AE Exposure Attribute Structure Pointer in WDR Mode	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetWDRExposureAttr*

4.4.2.5 CVI_ISP_SetAERouteAttr

【Description】

Set AE exposure allocation policy.

【Syntax】

```
CVI_S32 CVI_ISP_SetAERouteAttr(VI_PIPE ViPipe, const ISP_AE_ROUTE_S␣
↪*pstAERouteAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteAttr	AE Exposure Allocation Policy Structure Pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

This interface is used to set the AE exposure assignment route. The AE calculated exposure will be assigned according to the set route. Users can set the exposure priority, gain priority and aperture priority according to their needs.

The AE assignment route follows the following rules :

- A maximum of 16 nodes are supported, each of which has three components: exposure time, gain and aperture. The gain includes analog gain, digital gain and ISP digital gain.
- The exposure time unit in the node is us, which cannot be set to 0 or too small to cause the actual number of exposure rows to be 0, otherwise an exception may occur.

- The ring component only supports P-Iris, not DC-Iris, because DC-Iris cannot be precisely controlled, the DC-Iris and manual aperture lens aperture components are invalid.That is, when the aperture type is DC-Iris, the node aperture component has no effect on the exposure distribution.
- Node exposure is the product of exposure time, gain and aperture.

  Node exposure needs to increase monotonically.

  The exposure of the latter node should be greater than or equal to that of the previous node.

  The first node has the lowest exposure and the last node has the highest exposure.
- The exposure of adjacent nodes increases, then there should be one component increase, while the other components are fixed, and the increased component determines the allocation strategy for this segment of the route.For example, if the exposure time component increases, the strategy for assigning this segment of the route is to give priority to the exposure time.
- Setting equal exposure nodes is not supported.

Users can set different routes according to different scenarios, and assign routes to support dynamic switching.

- For DC-Iris and manual aperture lenses, the default AE allocation strategy is to allocate exposure time first, then gain.For P-Iris lenses, the

  default AE allocation strategy is to first adjust the aperture, set the aperture to the maximum, then adjust the exposure time, and finally allocate

  the gain.If the current exposure is not within the user-set route range, it is assigned by default policy.
- For DC-Iris and P-Iris switching online, AE route is reset to the default allocation policy that matches the aperture type, and the user can set AE

  route when switching the aperture type as needed.
- When the frame is automatically downscaled, the change in maximum exposure time is updated to the allocation route.
- When switching frame rates, if the user sets a maximum exposure target time greater than the maximum exposure time allowed by one frame after

  switching, the maximum exposure time of the allocation route will be updated to the maximum exposure time allowed by one frame after switching.
- When automatic frame dropping, linear and WDR mode switching, frame rate switching, limiting exposure time or maximum and minimum gain occur, the

  actual effective AE route may not be consistent with MPI settings, and the actual effective AE route can be obtained through

  CVI_ISP_QueryExposureInfo.

【Example】

```
//Set the route of gain 128x
VI_PIPE ViPipe = 0;

ISP_AE_ROUTE_S   stRoute;
CVI_ISP_SetAERouteAttr(ViPipe, &stRoute);

stRoute.u32TotalNum = 3;

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

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

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

CVI_ISP_SetAERouteAttr(ViPipe, &stRoute);
```

【Related Topic】

- *CVI_ISP_GetAERouteAttr*
- *ISP_AE_ROUTE_S*

4.4.2.6 CVI_ISP_GetAERouteAttr

【Description】

Gets the AE exposure allocation policy.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteAttr	AE Exposure Allocation Policy Structure Pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetAERouteAttr*

4.4.2.7 CVI_ISP_SetAERouteAttrEx

【Description】

Set AE exposure allocation extended properties to support setting sensor analog gain, sensor digital gain and ISP digital gain in AE allocation strategy respectively.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteAttrEx	AE Exposure Allocation Policy Extended Attribute Structure Pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- This interface is used to set the AE exposure assignment extension properties.
- The exposure calculated by AE will be assigned according to the set route.

  Users can set exposure time priority, sensor analog gain priority, sensor digital gain priority, ISP digital gain priority and aperture priority

  according to their needs.This interface can be used to set exposure assignment route in WDR mode, alleviate power flicker caused by multi-frame

  synthetic WDR in normal room illumination, and optimize image effect in WDR mode.
- Effectiveness of the AE exposure assignment extended property can be achieved by configuring bAERouteExValid in the CVI_ISP_SetExposureAttr

  interface.Use extended AE route when bAERouteExValid is CVI_TRUE, otherwise use normal AE route
- Other considerations are the same as the rules for *CVI_ISP_SetAERouteAttr*

【Example】

None.

【Related Topic】

- *CVI_ISP_GetAERouteAttrEx*
- *ISP_AE_ROUTE_EX_S*

4.4.2.8 CVI_ISP_GetAERouteAttrEx

【Description】

Gets the AE exposure allocation policy extended properties.

【Syntax】

```
CVI_S32 CVI_ISP_GetAERouteAttrEx(VI_PIPE ViPipe, ISP_AE_ROUTE_EX_S␣
↪*pstAERouteAttrEx);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteAttrEx	AE Exposure Allocation Policy Extended Attribute Structure Pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
VI_PIPE ViPipe = 0;

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

【Related Topic】

- *CVI_ISP_SetAERouteAttrEx*

4.4.2.9 CVI_ISP_QueryExposureInfo

【Descirption】

Obtain AE internal status information, including 256-segment histogram and average brightness statistics, as well as exposure time, gain, exposure and actual effective AE route information in AE operational status.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstExpInfo	Exposure internal state information structure pointer output	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

The exposure time is measured in microseconds(us), and the sensor analog gain, sensor digital gain, and ISP digital gain are obtained in multiples with a precision of 10 bits.

【Example】

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

CVI_ISP_QueryExposureInfo(ViPipe, &stExpInfo);

printf( “sensor used again = %d\n” , stExpInfo. u32AGain);
printf( “sensor used dgain = %d\n” , stExpInfo. u32DGain);
printf( “isp used dgain = %d\n” , stExpInfo. u32ISPDGain);
printf( “WDR Exposure ratio = %d\n” , stExpInfo. u32WDRExpRatio);
```

【Related Topic】

None.

4.4.2.10 CVI_ISP_SetAntiFlicker

【Descirption】

Set AE anti flicker function.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
enable	enable AE anti flicker	Input
frequency	Anti-flicker frequency	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
//Set AE anti flicker 50Hz

VI_PIPE ViPipe = 0;
CVI_U8 enable, frequency;
enable = 1;
frequency = 50;

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

【Related Topic】

- *CVI_ISP_GetAntiFlicker*

4.4.2.11 CVI_ISP_GetAntiFlicker

【Descirption】

Get AE anti flicker settings.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pEnable	Enable AE anti flicker	Input
pFrequency	Anti-flicker frequency	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

None.

4.4.2.12 CVI_ISP_QueryFps

【Descirption】

Get the current fps of AE.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pFps	AE fps pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_QueryExposureInfo*

4.4.2.13 CVI_ISP_GetCurrentLvX100

【Descirption】

It is used to obtain the light metering result of the current AE, that is, the ambient light brightness value LV. It can be used to judge night and day, and achieve the effect similar to a photoresistor.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
ps16Lv	Returns the current LV value	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
#include "cvi_ae.h"

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

bool checkDayOrNight(void)
{
    static bool dayOrNight = true;
    static CVI_U8 checkDayCount = 0;
```

(continues on next page)

(continued from previous page)

```
    static CVI_U8 checkNightCount = 0;
    CVI_S16 lv = 0;

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

int main(void)
{
    while (1) {
      sleep(1);
      printf("checkDayOrNight: %s\n", checkDayOrNight() ? "day" : "night");
    }
    return 0;
}
```

【Related Topic】

None.

4.4.2.14 CVI_ISP_SetSmartExposureAttr

【Descirption】

Set AE exposure properties in Smart mode. This parameter takes effect only when intelligent information is available.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstSmartExpAttr	AE exposure attribute structure pointer in smart mode	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

```
#include "cvi_ae.h"

VI_PIPE ViPipe = 0;
ISP_SMART_EXPOSURE_ATTR_S   stSmartExpAttr;

CVI_ISP_GetSmartExposureAttr(ViPipe, &stSmartExpAttr);

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

CVI_ISP_SetSmartExposureAttr(ViPipe, &stSmartExpAttr);
```

【Related Topic】

None.

4.4.2.15 CVI_ISP_GetSmartExposureAttr

【Descirption】

Gets the AE exposure properties in Smart mode. This parameter takes effect only when intelligent information is available.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstSmartExpAttr	AE exposure attribute structure pointer in smart mode	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetSmartExposureAttr*

4.4.2.16 CVI_ISP_SetAERouteSFAttr

【Descirption】

Set AE WDR short detection exposure allocation strategy.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteSFAttr	AE WDR short detection exposure allocation strategy structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetAERouteSFAttr*
- *ISP_AE_ROUTE_S*

4.4.2.17 CVI_ISP_GetAERouteSFAttr

【Descirption】

Get AE WDR short detection exposure allocation strategy.

【Syntax】

```
CVI_S32 CVI_ISP_GetAERouteSFAttr(VI_PIPE ViPipe, ISP_AE_ROUTE_S␣
↪*pstAERouteSFAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteSFAttr	AE WDR short detection exposure allocation strategy structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetAERouteSFAttr*

4.4.2.18 CVI_ISP_SetAERouteSFAttrEx

【Descirption】

Set AE WDR short detection exposure allocation extended attributes.

【Syntax】

```
CVI_S32 CVI_ISP_SetAERouteSFAttrEx(VI_PIPE ViPipe, const ISP_AE_ROUTE_S
↪*pstAERouteSFAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteSFAttr	AE WDR short detection exposure allocation strategy structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetAERouteSFAttrEx*
- *ISP_AE_ROUTE_EX_S*

4.4.2.19 CVI_ISP_GetAERouteSFAttrEx

【Descirption】

Get AE WDR short detection exposure allocation extended attributes.

【Syntax】

```
CVI_S32 CVI_ISP_GetAERouteSFAttrEx(VI_PIPE ViPipe, ISP_AE_ROUTE_S
↪*pstAERouteSFAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAERouteSFAttr	AE WDR short detection exposure allocation strategy structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetAERouteSFAttrEx*

4.4.3 Deep Learning Control Module

4.4.3.1 CVI_ISP_SetIrisAttr

【Descirption】

Set the control properties of the aperture.

This function can realize the setting of parameters such as manual aperture properties and aperture types.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	Aperture control attribute structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

- According to the actual aperture type of the connected lens, set the correct aperture type attribute, and then set the related DC-Iris/P-Iris control

  attribute.
- If it is connected with a manual iris lens, you can set the iris type to ISP_IRIS_DC_TYPE, and it is recommended to disable Deep Learning enablement at this

  time.
- The manual iris property is mainly used for debugging and can be set through this MPI.

  For P-Iris lenses, the manual enIrisFNO value will be affected by the maximum and minimum aperture target values.

  More parameters of the auto iris attribute need to be set by calling CVI_ISP_SetDcirisAttr and CVI_ISP_SetPirisAttr.

【Example】

None.

【Related Topic】

- *ISP_IRIS_ATTR_S*
- *CVI_ISP_SetDcirisAttr*
- *CVI_ISP_SetPirisAttr*

4.4.3.2 CVI_ISP_GetIrisAttr

【Descirption】

Gets the control properties of the aperture.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	Aperture control attribute structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

None.

4.4.3.3 CVI_ISP_SetDcirisAttr

【Descirption】

Set the control properties of the DC-Iris Deep Learning algorithm, this function can realize the parameter setting of the DC-Iris automatic iris.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	DC-Iris automatic aperture control property structure pointer.	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

The DC-Iris aperture control adopts the PID algorithm, which adjusts the PWM duty cycle to control the aperture size according to the screen brightness.

When the exposure time and gain reach the minimum target value, it will enter the aperture control area.

When the iris control can meet the requirements of the target brightness, AE returns directly, keeping the exposure time and gain unchanged.

When the brightness of the screen is stable and the PWM duty cycle is maintained at the open value for a period of time, the Deep Learning algorithm will consider that the aperture has been opened to the maximum, exit the aperture control area, and return the control to AE

【Example】

None.

【Related Topic】

- *ISP_IRIS_ATTR_S*
- *ISP_DCIRIS_ATTR_S*

4.4.3.4 CVI_ISP_GetDcirisAttr

【Descirption】

Gets the control properties of the DC-Iris automatic aperture.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	DC-Iris automatic aperture control property structure pointer.	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

None.

【Example】

None.

【Related Topic】

None.

4.4.3.5 CVI_ISP_SetPirisAttr

【Descirption】

Set the control properties of P-Iris automatic aperture

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	P-Iris auto iris control attribute structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

P-iris is not currently supported

【Example】

None.

【Related Topic】

- *ISP_PIRIS_ATTR_S*

4.4.3.6 CVI_ISP_GetPirisAttr

【Descirption】

Get the control properties of P-Iris automatic aperture

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstIrisAttr	P-Iris auto iris control attribute structure pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_ae.h
- Library files: libae.a

【Note】

P-iris is not currently supported

【Example】

None.

【Related Topic】

- *ISP_PIRIS_ATTR_S*

4.5 Data Types

4.5.1 Register

- *AE_SENSOR_REGISTER_S* : Define sensor registration structure.
- *AE_SENSOR_EXP_FUNC_S* : Define the sensor callback function structure.
- *AE_SENSOR_DEFAULT_S* : Defines the initialization parameter structure of the AE algorithm library.
- *AE_ACCURACY_E* : Enumeration of precision types defining exposure time and gain.
- *AE_ACCURACY_S* : A structure that defines the precision of exposure time and gain.

4.5.1.1 AE_SENSOR_REGISTER_S

【Description】

Define the sensor registration structure.

【Syntax】

```
typedef struct cviAE_SENSOR_REGISTER_S

{

  AE_SENSOR_EXP_FUNC_S stSnsExp;

} AE_SENSOR_REGISTER_S;
```

【Member】

Member	Description
stSnsExp	Callback function structure registered by sensor

【Note】

None.

【Related Data Type and Interface】

- *AE_SENSOR_EXP_FUNC_S*

4.5.1.2 AE_SENSOR_EXP_FUNC_S

【Description】

Define the sensor callback function structure.

【Syntax】

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

【Member】

Member	Description
pfn_cmos_get_ae_default	Gets the callback function pointer of the initial value of the AE algorithm library.
pfn_cmos_fps_set	Set the frame rate of sensor.
pfn_cmos_slow_framerate_set	Set the sensor down frame.
pfn_cmos_inttime_update	Set the exposure time of sensor.
pfn_cmos_gains_update	Set the analog gain and digital gain of sensor.
pfn_cmos_again_calc_table	Calculate the analog gain of TABLE type sensor.
pfn_cmos_dgain_calc_table	Calculate the digital gain of TABLE type sensor.
pfn_cmos_get_inttime_max	In WDR mode, the pointer of callback function to calculate the maximum exposure time of short frame is strongly related to sensor.
pfn_cmos_ae_fswdr_attr_set	In 2to1LineWDR mode, set the long frame mode

【Note】

- If the callback function pointer does not need to be assigned, it needs to be set to NULL.
- For information on how to convert the exposure time and gain set in pfn_cmos_inttime_update and pfn_cmos_gains_update into sensor configuration

  value and sensor strength, please refer to the sensor manual.

【Related Data Type and Interface】

- *AE_SENSOR_DEFAULT_S*
- ISP_SENSOR_EXP_FUNC_S

4.5.1.3 AE_SENSOR_DEFAULT_S

【Description】

Define the initialization parameter structure of AE algorithm library.

【Syntax】

```
typedef struct _AE_SENSOR_DEFAULT_S
{
  CVI_U8 au8HistThresh[HIST_THRESH_NUM];
  CVI_U8 u8AeCompensation;
  CVI_U32 u32LinesPer500ms;
  CVI_U32 u32FlickerFreq;
  CVI_U32 u32HmaxTimes;
  CVI_U32 u32InitExposure;
  CVI_U32 u32InitAESpeed;
  CVI_U32 u32InitAETolerance;
  CVI_U32 u32FullLinesStd;
  CVI_U32 u32FullLinesMax;
  CVI_U32 u32FullLines;
  CVI_U32 u32MaxIntTime;
  CVI_U32 u32MinIntTime;
```

(continues on next page)

(continued from previous page)

```
  CVI_U32 u32MaxIntTimeTarget;
  CVI_U32 u32MinIntTimeTarget;
  AE_ACCURACY_S stIntTimeAccu;
  CVI_U32 u32MaxAgain;
  CVI_U32 u32MinAgain;
  CVI_U32 u32MaxAgainTarget;
  CVI_U32 u32MinAgainTarget;
  AE_ACCURACY_S stAgainAccu;
  CVI_U32 u32MaxDgain;
  CVI_U32 u32MinDgain;
  CVI_U32 u32MaxDgainTarget;
  CVI_U32 u32MinDgainTarget;
  AE_ACCURACY_S stDgainAccu;
  CVI_U32 u32MaxISPDgainTarget;
  CVI_U32 u32MinISPDgainTarget;
  CVI_U32 u32ISPDgainShift;
  CVI_U32 u32MaxIntTimeStep;
  CVI_U32 u32LFMaxShortTime;
  CVI_U32 u32LFMinExposure;
  ISP_AE_STRATEGY_E enAeExpMode;
  CVI_U16 u16ISOCalCoef;
  CVI_U8 u8AERunInterval;
  CVI_FLOAT f32Fps;
  CVI_FLOAT f32MinFps;
  CVI_U32 denom;
  CVI_U32 u32AEResponseFrame;
  CVI_U32 u32SnsStableFrame;
  AE_BLC_TYPE_E enBlcType;
  ISP_SNS_GAIN_MODE_E    enWDRGainMode;
} AE_SENSOR_DEFAULT_S;
```

【Member】

Member	Description
u8AeCompensation	AE brightness target value, the value range is [0255], it is recommended to use 0x38 ~ 0x40.
u32LinesPer500ms	Total number of rows per 500ms.
u32FlickerFreq	The anti-flash frequency is 256 times of the current power frequency.
u32HmaxTimes	Time of reading a line by sensor, unit: ns.
u32InitExposure	The default initial exposure is equal to the exposure time (lines) * system gain (6bit decimal precision). AE algorithm uses this value as the exposure of the initial 5 frames
u32InitAESpeed	The default initial AE adjustment speed is used as the adjustment speed of the initial 100 frames
u32InitAETolerance	The AE algorithm uses this value as the initial 100 frame exposure tolerance value, which can be used to set the brightness range of the first AE convergence stability flag

continues on next page

Table 4.1 – continued from previous page

Member	Description
u32FullLinesStd	The effective number of lines per frame at the base frame rate.
u32FullLinesMax	The maximum number of rows that can be reached in one frame after sensor frame dropping. It is generally set to the maximum number of rows supported by the sensor.
u32FullLines	The total number of effective rows per sensor frame. When using CVI AE algorithm, when calling back cmos_fps_set and cmos_slow_framerate_set, this value must be assigned to return the total number of effective rows per frame.
u32MaxIntTime	Maximum exposure time, in rows.
u32MinIntTime	Minimum exposure time, in rows.
u32MaxIntTimeTarget	Maximum exposure time target value, in rows.
u32MinIntTimeTarget	Minimum exposure time target value, in rows.。
stIntTimeAccu	Exposure time accuracy.
u32MaxAgain	Maximum analog gain in multiples.
u32MinAgain	Minimum analog gain in multiples.
u32MaxAgainTarget	Maximum analog gain target value, in multiples.
u32MinAgainTarget	Minimum analog gain target value, in multiples.
stAgainAccu	Simulation gain accuracy.
u32MaxDgain	Maximum digital gain in multiples.
u32MinDgain	Minimum digital gain in multiples.
u32MaxDgainTarget	Maximum digital gain target value, in multiples.
u32MinDgainTarget	Minimum digital gain target value, in multiples.
stDgainAccu	Digital gain accuracy.
u32MaxISPDgainTarget	Maximum ISP digital gain target value, in multiples.
u32MinISPDgainTarget	Minimum ISP digital gain target value, in multiples.
u32ISPDgainShift	ISP digital gain accuracy.
u32MaxIntTimeStep	The maximum adjustment step of short frame exposure time reduction, in rows. Valid only in two frame synthesis mode.
u32LFMaxShortTime	Maximum exposure time of short frame in automatic long frame mode.
u32LFMinExposure	Minimum exposure time of short frame in automatic long frame mode.
enAeExpMode	Default exposure strategy, high light priority or low light priority. It is recommended that FSWDR mode be set to low light priority, linear mode and BuiltIn WDR be set to high light priority. If this value is not set, highlight takes precedence by default.
u16ISOCalCoef	ISO calibration coefficient is used to ensure that the ISO displayed in the DCF information required for photographing is standard, with an accuracy of 8bit. Value range: [0x0, 0xFFFF], default value is 0x100.
u8AERunInterval	The default AE algorithm running interval, in frames. If it is not set, AE is executed once per frame by default. Value range: (0x0, 0xFF]
f32Fps	Reference frame rate

continues on next page

Table 4.1 – continued from previous page

Member	Description
u32AEResponseFrame	The number of frames required for sensor exposure time/gain synchronization to take effect

【Note】

- When the linear/WDR mode is switched, the pfn_cmos_get_ae_default function will be called back to update the AE related default parameters.

  If the WDR mode needs to use AE extended route assignment, but linear mode does not, it is recommended to clear AE route in cmos_get_ae_default

  function first: bAERouteExValid = CVI_FALSE, stAERouteAttr.u32TotalNum= 0, stAERouteAttrEx.u32TotalNum = 0, and then assign values to the WDR branch as needed.

- u32LFMaxShortTime is the maximum exposure time of short frame in automatic long frame mode.

  If this parameter is set too small, the noise performance of bright area will be worse in automatic long frame mode.

【Related Data Type and Interface】

- ISP_SENSOR_EXP_FUNC_S

4.5.1.4 AE_ACCURACY_E

【Description】

Enumeration of precision types defining exposure time and gain

【Syntax】

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

【Member】

Member	Description
AE_ACCURACY_DB DB	Precision type.
AE_ACCURACY_LINEAR	Linear precision type.
AE_ACCURACY_TABLE	Form type.

【Note】

None.

【Related Data Type and Interface】

- *AE_ACCURACY_S*

4.5.1.5 AE_ACCURACY_S

【Description】

A structure that defines the precision of exposure time and gain.

【Syntax】

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

【Member】

Member	Description
enAccuType	Precision type, including linear type, DB type and TABLE type. .
f32Accuracy	Precision value.
f32Offset	The offset of the exposure time, supports positive and negative offset settings, in units of rows, this value configuration is strongly related to the sensor.

【Note】

None.

【Related Data Type and Interface】

None.

4.5.2 AE

- *ISP_AE_MODE_E* :Define the mode of auto exposure.
- *ISP_AE_STRATEGY_E* :Define AE exposure strategy mode.
- *ISP_AE_DELAY_S* :Define AE delay attributes.
- *ISP_AE_RANGE_S* :Define the maximum and minimum exposure time or gain.
- *ISP_ANTIFLICKER_MODE_E* :Define the anti-flash mode.
- *ISP_ANTIFLICKER_S* :Define the anti-flash attributes.
- *ISP_SUBFLICKER_S* :Define the sub anti flicker attribute of ISP image.
- *ISP_FSWDR_MODE_E* :Define ISP FSWDR operation mode.
- *ISP_AE_ATTR_S* :Define auto exposure attributes.
- *ISP_ME_ATTR_S* :Define manual exposure attributes.
- *ISP_EXPOSURE_ATTR_S* :Define ISP exposure attributes.

- *ISP_WDR_EXPOSURE_ATTR_S* :Define exposure attributes in WDR mode.
- *ISP_AE_ROUTE_NODE_S* :Define AE assignment route node attributes.
- *ISP_AE_ROUTE_S* :Define AE exposure assignment policy attributes.
- *ISP_AE_ROUTE_EX_NODE_S* :Define the attribute of AE extension assignment route node.
- *ISP_AE_ROUTE_EX_S* :Define extended attributes of AE exposure allocation strategy.
- *ISP_EXP_INFO_S* :Define the internal state information of ISP exposure.

4.5.2.1 ISP_AE_MODE_E

【Description】

Define the mode of auto exposure.

【Syntax】

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

【Member】

Member	Description
AE_MODE_SLOW_SHUTTER	Automatic frame down mode, namely SLOW_SHUTTER mode.
AE_MODE_FIX_FRAME_RATE	Fixed frame rate mode.

【Note】

- Auto frame down mode means that the exposure time will be increased first to reduce the gain.

  When the sensor gain reaches the maximum value set by the user, the auto exposure adjustment will gradually reduce the frame rate and extend the

  exposure time until the exposure time is equal to the maximum time of auto exposure.

  In low illumination environment, the noise is small, but the frame rate will be reduced.

- The fixed frame rate mode is to keep the frame rate unchanged in the automatic exposure adjustment, and the noise will be greater in the low

  illumination environment.

【Related Data Type and Interface】

None.

4.5.2.2 ISP_AE_STRATEGY_E

【Description】

Define AE exposure strategy mode.

【Syntax】

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

【Member】

Member	Description
AE_EXP_HIGHLIGHT_PRIOR	High light priority exposure mode.
AE_EXP_LOWLIGHT_PRIOR	High light priority exposure mode.

【Note】

- In linear mode, the AE algorithm' s default exposure strategy is highlight first to avoid overexposure, but in backlight scene, the brightness is low in the dark.

  At this time, to focus on the dark area, low light priority mode can be adopted, but the bright area will be overexposed easily.

- In WDR mode, AE algorithm uses low light priority mode by default to control long frame exposure.

  At this time, manual exposure time, maximum and minimum exposure time, AE_Route, and AE_Route_Ex exposure time parameters affect long frame exposure time.

  If it is switched to high light priority mode, short frame exposure is controlled.

  At this time, manual exposure time, maximum and minimum exposure time, AE_Route, and AE_Route_Ex exposure time parameters affect short frame exposure time.

【Related Data Type and Interface】

None.

4.5.2.3 ISP_AE_DELAY_S

【Description】

Define AE delay attributes.

【Syntax】

```
typedef struct _ISP_AE_DELAY_S
{
  CVI_U16 u16BlackDelayFrame;
```

(continues on next page)

(continued from previous page)

```
  CVI_U16 u16WhiteDelayFrame;
} ISP_AE_DELAY_S;
```

【Member】

Member	Description
u16BlackDelayFrame	When the image brightness is less than the target brightness for more than u16BlackDelayFrame frames, AE starts to adjust.
u16WhiteDelayFrame	When the image brightness exceeds the target brightness for more than u16WhiteDelayFrame frames, AE starts to adjust.

【Note】

- The larger the u16BlackDelayFrame/u16WhiteDelayFrame setting is, the longer it takes to adjust to the target brightness with the same AE adjustment step.

  Human eyes are sensitive to overexposure.

  It is recommended that u16WhiteDelayFrame be set smaller than u16BlackDelayFrame.

- If the frame rate is low, it is recommended that u16BlackDelayFrame/u16WhiteDelayFrame should not be set too large, otherwise the AE convergence time will be longer.

【Related Data Type and Interface】

None.

4.5.2.4 ISP_AE_RANGE_S

【Description】

Define the maximum and minimum exposure time or gain.

【Syntax】

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

【Member】

Member	Description
u32Max	Maximum value
u32Min	Minimum Value

【Note】

The minimum value cannot be greater than the maximum value.

【Related Data Type and Interface】

None.

4.5.2.5 ISP_ANTIFLICKER_MODE_E

【Description】

Define the anti-flash mode.

【Syntax】

```
typedef enum cviISP_ANTIFLICKER_MODE_E

{

  ISP_ANTIFLICKER_NORMAL_MODE = 0x0,

  ISP_ANTIFLICKER_AUTO_MODE = 0x1,

  ISP_ANTIFLICKER_MODE_BUTT

} ISP_ANTIFLICKER_MODE_E;
```

【Member】

Member	Description
ISP_ANTIFLICKER_NORMAL_MODE	Normal anti-flash mode.
ISP_ANTIFLICKER_AUTO_MODE	Automatic anti-flash mode.

【Note】

- ISP_ANTIFLICKER_NORMAL_MODE for ordinary flash mode, exposure time can be adjusted according to the brightness, exposure time fixed for the smallest

  1/120 SEC (60 hz) or 1/100 SEC (50 hz) and is not restricted by exposure time minimum value.
- In an environment with lights: The exposure time matches the frequency of the light source, preventing image flickering.
- High brightness environment: The higher the brightness, the shorter the exposure time.

  The minimum exposure time of normal anti-flash mode is not adjusted for high brightness, so overexposure can occur.
- ISP_ANTIFLICKER_AUTO_MODE resistance for automatic flash mode, exposure time can be adjusted according to the brightness, minimum exposure time can

  achieve the minimum exposure time sensor.

  The difference with the ordinary anti-flash mode is mainly reflected in the high brightness environment.
- High brightness environment: The minimum exposure time can reach the minimum exposure time of the sensor, effectively suppressing overexposure, but

  the anti-flash failure in this case.

【Related Data Type and Interface】

None.

4.5.2.6 ISP_ANTIFLICKER_S

【Description】

Define the anti-flash attributes.

【Syntax】

```
typedef struct cviISP_ANTIFLICKER_S

{

  CVI_BOOL bEnable;

  CVI_U8 u8Frequency;

ISP_ANTIFLICKER_MODE_E enMode;

} ISP_ANTIFLICKER_S;
```

【Member】

Member	Description
bEnable	When bEnable is CVI_TRUE, image anti-flash is enabled, and when it is CVI_FALSE, image anti-flash is not enabled.
enFrequency	Anti-flicker frequency value. Value range: [0, 1], the default value is 0. 0: 60Hz 1: 50Hz
enMode	Anti-flash mode, normal anti-flash or automatic anti-flash.

【Note】

The exposure time after anti-flashing is enabled will be limited by the maximum/minimum exposure time.

If the minimum anti-flashing time is greater than the maximum exposure time, the exposure time after anti-flashing is enabled will be limited to the maximum exposure time.

【Related Data Type and Interface】

- *ISP_ANTIFLICKER_MODE_E*

4.5.2.7 ISP_SUBFLICKER_S

【Description】

Define the sub anti flicker attribute of ISP image.

【Syntax】

```
typedef struct cviISP_SUBFLICKER_S

{

  CVI_BOOL bEnable;

  CVI_U8 u8LumaDiff;

} ISP_SUBFLICKER_S;
```

【Member】

Member	Description
bEnable	When bEnable is CVI_TRUE, the image sub-anti-flash function is enabled, and when it is CVI_FALSE, the image sub-anti-flash function is not enabled.
u8LumaDiff	Anti-flash setting, range [0x0, 0x64]. When the sub-anti-flash function is in effect, the larger the value is, the closer it is to anti-flash.

【Note】

- Force resisting flash mode, the minimum exposure time fixed for the 1/120 SEC (60 hz) or 1/100 SEC (50 hz), in some scenarios, such as indoor

  targeting the backlit scenes outside the window) images may be blown severely, but not to flash flash power frequency and the better picture.

  In this case, the subantiflash mode is introduced to achieve a balance between overexposure and flicker.

  Under the mandatory anti-flicker mode, if the image brightness is less than (AeCompensation + u8LumaDiff) when the sub-anti-flicker function takes

  effect, the exposure time will still be fixed as 1/120 sec(60Hz) or 1/100 sec(50Hz), the minimum anti-flicker time, to prevent the image flicker.

  If the screen brightness is higher than (AeCompensation + u8LumaDiff), cancel the anti-flicker and adjust the target brightness of the picture to (AeCompensation + u8LumaDiff), and introduce a certain degree of flicker to avoid serious overexposure of the picture.

- only in open flash, forced resisting flash mode and under the premise of flash frequency resistance value is not equal to 0, the flash function to

  take effect.

【Related Data Type and Interface】

None.

4.5.2.8 ISP_FSWDR_MODE_E

【Description】

Define ISP FSWDR operation mode.

【Syntax】

```
typedef enum cviISP_FSWDR_MODE_E

{

  ISP_FSWDR_NORMAL_MODE = 0x0,

  ISP_FSWDR_LONG_FRAME_MODE = 0x1,

  ISP_FSWDR_AUTO_LONG_FRAME_MODE = 0x2,

  ISP_FSWDR_MODE_BUTT

}ISP_FSWDR_MODE_E;
```

【Member】

Member	Description
ISP_FSWDR_NORMAL_MODE	normal FSWDR mode, the AE and composition modules work according to the automatic/manual exposure ratio.
ISP_FSWDR_LONG_FRAME_MODE	Long frame mode. At this time, AE sets the short frame exposure time to the minimum value, and the long frame exposure time is close to the maximum value allowed by 1 frame.
ISP_FSWDR_AUTO_LONG_FRAME_MODE	**Automatic long frame mode, at this time, the AE short frame exposure time** limited by the sensor. When the exposure exceeds the threshold set by the sensor, it will automatically switch to long frame mode, and the synthesis module will only output long frame data.

【Note】

- Long frame mode only WDR mode effective line, automatic line long frame mode in only 2 to1 WDR mode effectively, and to ensure the quality of image

  and WDR module fusion threshold of motion detection and the length of the frame are automatically configured, does not support manual configuration.

  Switch the mode to WDR mode online.

  The default mode is normal FSWDR mode.

- Automatic long frame mode, manual exposure than to take effect, if manual exposure is greater than 1, even if the exposure is greater than the set threshold will not enter the long frame mode; If the manual exposure ratio is 1, it will automatically enter long frame mode.

【Related Data Type and Interface】

None.

4.5.2.9 ISP_AE_GAIN_TYPE_E

【Description】

Define how the gain is used.

【Syntax】

```
typedef enum _ISP_AE_GAIN_TYPE_E {

  AE_TYPE_GAIN = 0,

  AE_TYPE_ISO = 1,

  AE_TYPE_BUTT

} ISP_AE_GAIN_TYPE_E;
```

【Member】

Member	Description
enGainType	0 : use the gain method 1 : The way to use ISO num

【Note】

None.

4.5.2.10 ISP_AE_ATTR_S

【Description】

Define auto exposure attributes.

【Syntax】

```
typedef struct _ISP_AE_ATTR_S

{

 ISP_AE_RANGE_S stExpTimeRange;

 ISP_AE_RANGE_S stAGainRange;

 ISP_AE_RANGE_S stDGainRange;

```

(continues on next page)

(continued from previous page)

```
ISP_AE_RANGE_S stISPDGainRange;

ISP_AE_RANGE_S stSysGainRange;

CVI_U32 u32GainThreshold;

CVI_U8 u8Speed;

CVI_U16 u16BlackSpeedBias;

CVI_U8 u8Tolerance;

CVI_U8 u8Compensation;

CVI_U16 u16EVBias;

ISP_AE_STRATEGY_E enAEStrategyMode;

CVI_U16 u16HistRatioSlope;

CVI_U8 u8MaxHistOffset;

ISP_AE_MODE_E enAEMode;

ISP_ANTIFLICKER_S stAntiflicker;

ISP_SUBFLICKER_S stSubflicker;

ISP_AE_DELAY_S stAEDelayAttr;

CVI_BOOL bManualExpValue;

CVI_U32 u32ExpValue;

ISP_FSWDR_MODE_E enFSWDRMode;

CVI_BOOL bWDRQuick;

CVI_U16 u16ISOCalCoef;

ISP_AE_GAIN_TYPE_E enGainType;

ISP_AE_RANGE_S stISONumRange;

CVI_S16 s16IRCutOnLv;

CVI_S16 s16IRCutOffLv;
```

(continues on next page)

(continued from previous page)

```
  ISP_AE_IR_CUT_FORCE_STATUS enIRCutStatus;

  CVI_U8 au8AdjustTargetMin[LV_TOTAL_NUM];

  CVI_U8 au8AdjustTargetMax[LV_TOTAL_NUM];

  CVI_U16 u16LowBinThr;

  CVI_U16 u16HighBinThr;

  CVI_BOOL bEnableFaceAE;

  CVI_U8 u8FaceTargetLuma;

  CVI_U8 u8FaceWeight;

  CVI_U8 u8GridBvWeight;

  CVI_U32 au32Reserve[RESERVE_SIZE];

  CVI_U8 u8HighLightLumaThr;

  CVI_U8 u8HighLightBufLumaThr;

  CVI_U8 u8LowLightLumaThr;

  CVI_U8 u8LowLightBufLumaThr;

  CVI_BOOL bHistogramAssist;

} ISP_AE_ATTR_S;
```

【Member】

Member	Description
stExpTimeRange	Exposure time range, set the maximum and minimum values in microseconds (us). Value range: [0x0, 0xFFFFFFFF], The specific range is related to the sensor.
stISONumRange	ISO num range, set the maximum and minimum values, this setting is only valid when enGainType = AE_TYPE_ISO. Value range: [0x64, 0xFFFFFFFF], The specific range is related to the sensor.
stAGainRange	Sensor analog gain range, set maximum and minimum values, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF], The specific range is related to the sensor.
stDGainRange	Sensor digital gain range, set maximum and minimum values, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF], The specific range is related to the sensor.

continues on next page

Table 4.2 – continued from previous page

Member	Description
stISPDGainRange	ISP digital gain range, set maximum and minimum values, 10bit decimal precision. Value range: [0x400, 0x7FFF].
stSysGainRange	System gain range, set maximum and minimum values, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF], The specific range is related to the sensor.
u32GainThreshold	System gain threshold for automatic frame reduction, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF], The default value is 0x400000
u8Speed	The speed at which auto exposure is adjusted. Value range: [0x0, 0xFF], The default value is 0x40.
u16BlackSpeedBias	Screen from dark to bright AE adjusts the deviation of the speed, the larger the value, the faster the screen from dark to bright. Value range: [0x0, 0xFFFF], The default value is 0x90.
u8Tolerance	Tolerance deviation of screen brightness during automatic exposure adjustment. Value range: [0x0, 0xFF], The default value is 0x2.
u8Compensation	Target brightness during automatic exposure adjustment. Value range: [0x0, 0xFF], The default value is 0x38.
u16EVBias	Exposure deviation value during automatic exposure adjustment, 10bit decimal precision. Value range: [0x0, 0xFFFF], The default value is 0x400.
enAEStrategyMode	Automatic exposure strategy, high light priority or low light priority, the default value is AE_EXP_HIGHLIGHT_PRIOR
u16HistRatioSlope	When high/low light is prioritized, adjust the step size of the target brightness. Value range: [0x0, 0xFFFF], The default value is 0x8.
u8MaxHistOffset	The maximum extent to which the region of interest contributes to the statistical mean. Value range: [0x0, 0xFF], The default value is 0x10.
enAEMode	Auto exposure mode, auto drop frame mode or fixed frame rate mode. The default value is AE_MODE_FIX_FRAME_RATE
stAntiflicker	Anti-flash property settings. Anti-flash is disabled by default.
stSubflicker	Sub-anti-flash property setting. By default, sub-anti-flash is not enabled.
stAEDelayAttr	Delay property setting. Default u16BlackDelayFrame=0 u16WhiteDelayFrame=0.
bManualExpValue	Manual exposure is enabled, when the value is CVI_TRUE, the AE algorithm uses u32ExpValue as the current exposure to allocate exposure time and gain, and when it is CVI_FALSE, it uses the automatically calculated exposure to allocate. The default value is CVI_FALSE.
u32ExpValue	Manual exposure value, equal to exposure time (lines) * system gain (6bit decimal precision). The default value is 0 Value range: (0x0, 0xFFFFFFFF].

continues on next page

Table 4.2 – continued from previous page

Member	Description
enFSWDRMode	FSWDR mode of operation. The default is ISP_FSWDR_NORMAL_MODE.
bWDRQuick	In WDR mode, when the AE algorithm is readjusted from a stable state (the brightness error is less than or equal to the tolerance value u8Tolerance), the default adjustment of the first 50 frames will perform temporal filtering to make the adjustment smoother. When the value is CVI_TRUE, the 50-frame temporal filtering is canceled to make AE converge faster. Defaults to CVI_FALSE.
u16ISOCalCoef	The ISO calibration coefficient is used to ensure that the ISO displayed in the DCF information required for taking pictures is standard and 8bit precision. Value range: [0x0, 0xFFFF], The default value is 0x100.
enGainType	use way of gain. Choose to control the gain by ISO Num or Gain.
au8AdjustTargetMin	Sets the minimum value of the target brightness for each LV AE of ambient brightness to converge. Value range: (0x0, 0x100) The default value of LV -5 ~ 15 is { 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 45, 50, 50, 50, 50, 50, 50, 60, 60, 60}
au8AdjustTargetMax	Set the maximum value of the target luminance where each LV AE of ambient luminance converges. Value range: (0x0, 0x100) The default value of LV -5 ~ 15 is { 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 60, 60, 60, 60, 60, 60, 60, 70, 70, 70}
u16LowBinThr	When the statistical value of the AE window is lower than lowBinThr(8 bits) and the cumulative number of windows is lower than 25% of the total number of windows, AE does not use the statistical value of this window to measure light. The default is 10
u16HighBinThr	When the statistical value of the AE window is higher than HighBinThr(8 bits) and the cumulative number of windows is lower than 10% of the total number of windows, AE does not use the statistical value of this window to measure light. The default is 256
s16IRCutOnLv	When IR Cut is switchable, use the ambient Lv value of IR Cut, the precision is 100 threshold, The default value is 0
s16IRCutOffLv	When the IR Cut is switchable, the Lv value of the environment without the IR Cut has an accuracy of 100 The default value is 700
enIRCutStatus	When IR Cut can be switched, the mode setting of IR Cut switching, the default value is AE_IR_CUT_FORCE_AUTO
bEnableFaceAE	Face recognition linked AE metering (Face AE) enabled
u8FaceTargetLuma	Target brightness for face metering (8 bits)
u8FaceWeight	The weight ratio of face metering and overall environment metering

continues on next page

Table 4.2 – continued from previous page

Member	Description
u8GridBvWeight	AE metering is divided into two types: block luma / bvStep average. The default is luma-based. This parameter can be used to set the metering weight of luma-based metering mixed with bvStep. Highlight priority. If the image is too dark, you can increase it appropriately This parameter weight
au32Reserve[RESERVE_SIZE]	Reserved field, meaningless
u8HighLightLumaThr	The brightness threshold of the highlighted area
u8HighLightBufLumaThr	Brightness threshold of the highlight buffer
u8LowLightLumaThr	Brightness threshold for low light areas
u8LowLightBufLumaThr	Brightness threshold for low light buffers
bHistogramAssist	Use histogram statistics to assist the highlight strategy to deal with the case where the highlight area is smaller than a block size

【Note】

- Automatic mode, change the manual exposure attribute’ s value will not take effect.
- The maximum and minimum time and gain of automatic exposure.
- The exposure time and gain can be limited according to different scenes.

  If there is a high-speed moving object scene, the maximum exposure time can be limited to a small value, which can reduce the dragging phenomenon of

  moving objects.
- System gain for automatic exposure
- If (minimum sensor analog gain * minimum sensor digital gain * minimum ISP digital gain) is less than the minimum system gain, the minimum gain in

  the internal calculation of AE algorithm will be limited to the minimum system gain.
- If (maximum analog gain of sensor * maximum digital gain of sensor * maximum digital gain of ISP) is greater than maximum system gain, the maximum

  gain in the internal calculation of AE algorithm will be limited to the maximum system gain.
- It is recommended to set the maximum and minimum system gain to limit the analog gain of the sensor, the digital gain of the sensor, and the ISP

  digital gain respectively.
- If the ISP digital gain of higher precision is limited to 1 times, flicker may occur.
- In fact, the AE algorithm internally uses the system gain to calculate the maximum/minimum exposure, rather than directly limiting the value of a gain.
- For example, the maximum system gain is limited to 1 times, but the minimum analog gain of the sensor is limited to 2 times.

  Then the actual effective results are subject to the limits of the analog gain of the sensor, and the same applies to the limits of the digital gain

  of the sensor and the digital gain of the ISP.

System gain threshold for automatic frame drop

In SLOW_SHUTTER mode, when the system gain reaches a set threshold, the system automatically enters the SLOW_SHUTTER mode.

- u8Speed convergence speed for setting automatic exposure, the value is, the greater the exposure of the faster convergence speed, but also leads to repeatedly appeared in the process of convergence oscillation.
- u16BlackSpeedBias used to set from dark to bright picture AE to adjust speed deviation. Under default u8Speed, the screen speed from light to dark will be faster than the screen speed from dark to light. If the adjustment speed of AE in both directions is similar, you can adjust u16BlackSpeedBias.
- brightness compensation property u8Compensation targets to adjust the exposure intensity. The larger the exposure brightness compensation value, the higher the image brightness.
- exposure deviation property u16EVBias targets to under a special scene images to brightness fine-tuning, also can be regarded as the brightness of the higher precision compensation value, by adjusting the value to change the picture target brightness, the real effect of AE target brightness for u8Compensation * u16EVBias / 1024. The larger the u8Compensation value is, the higher the image brightness will be.
- exposure tolerance deviation property u8Tolerance used to adjust the exposure to the environment, the sensitivity of the tolerance deviation, the greater the exposure, the less sensitive, and may lead to the same target brightness values repeatedly adjust brightness difference is, the greater the so recommend this property cannot be set too big.
- enAEStrategyMode exposure strategy attribute is used to select the high priority or low light exposure policy priority. Highlighting first means being sensitive to highlights and trying to avoid overexposure. Low light priority means being sensitive to low light and trying to see dark areas clearly, whether or not the image is overexposed. The default exposure strategy is highlight first, The user can adjust it according to the needs of the scene. When the highlight is priority, the user can set u8HighLightLumaThr to determine the brightness threshold of the highlight area. When there is windows exceeding this brightness threshold in the screen (the weight of the windows is not 0), When the brightness of all windows is lower than u8HighLightBufLumaThr, the original target brightness will be restored. When the brightness of low light is priority, If there is windows in the screen below the u8LowLightLumaThr brightness threshold (the weight of this windows is not 0), Then the target brightness would be increased to brighten the brightness of the low-light windows. When the brightness of all windows was greater than u8LowLightBufLumaThr, the original target brightness would be restored.
- u16HistRatioSlope is used to set the weight of interested area. If the highlight priority mode is used, this value sets the weight of the highlight area. The larger the value, the more sensitive it is to the highlight area. On the other hand, in low-light priority mode, this value sets the weight of the low-light area. A higher value means more sensitive to the dark areas. It is recommended that u16HistRatioSlope not exceed 0x100.
- automatic exposure, can be set on the impact of interested area on statistical average u8MaxHistOffset to the greatest extent. This value limits how much weight can be added to raise u16HistRatioSlope. A value of 0, no matter how large the u16HistRatioSlope is, will not give special treatment to the highlighted or low-light areas, and the statistical mean will be the original value. Reasonable setting of this value can ensure that the average brightness of any scene after AE stabilization is within a certain range. In the high-priority

exposure mode, if this value is set to a large extent, the overall brightness of the picture may be low in the scene with slightly higher contrast, such as the sunny outdoor scene with sky and trees, because the effect of bright area sky is guaranteed first. This problem can be solved by limiting the value and thus the degree of tilt to bright areas.

- exposure switching strategy, it is best update u16HistRatioSlope and u8MaxHistOffset value at the same time, otherwise the two values will adopt a strategy of configuration, may not correspond with the expected effect.
- when doing light inhibition, it is suggested that the specular preferred exposure mode, by reducing the AE target brightness, at the same time, setting up reasonable u16HistRatioSlope and u8MaxHistOffset to suppress strong light, dark space will be enabled to DRC to see clearly. Low light priority can be used to realize the backlight compensation of non-specified areas.
- AE for automatic exposure control type, can be set enAEMode exposure mode. This value can be set to SLOW_SHUTTER or fixed frame rate mode. Slow shutter mode is usually used to automatically drop frames in low-light scenes to reduce picture noise.
- flash attribute structure stAntiflicker resistance can be used to set flash enabled, flash frequency and resisting properties such as flash mode. In FSWDR mode, anti-flicker is effective only for long frames. Since the maximum exposure time for short frames is limited as exposure ratio changes, it is recommended to use automatic anti-flash mode for FSWDR mode.
- the flash attribute structure stSubflicker resistance can be used to set the degree of flash flash enabled and resisting properties. If there is an automatic aperture, it is recommended to turn off the sub-anti-flash function.
- AE for automatic exposure control type, can be set structure stAEDelayAttr AE delay effect attributes. The reasonable setting of this value can improve the stability of the picture brightness and prevent the change of the picture brightness caused by fast moving objects. This value should be increased appropriately at low bit rate Settings to avoid blocking effect when AE is adjusted.
- users can will be for CVI_TRUE bManualExpValue, manually exposure u32ExpValue to block out the AE exposure to adjust part of the algorithm, only to exposure distribution part. The value of u32ExpValue is limited by the maximum and minimum exposure. The maximum and minimum exposures correspond to the product of the maximum and minimum exposure time and gain, respectively.
- FSWDR operation mode change, need to modify the cmos. Corresponding callback function in c: cmos_get_ae_default, cmos_fps_set, cmos_get_inttime_max, to ensure the long and short exposure time frame in a reasonable range. When switching between normal WDR and long-frame modes, the cmos_get_ae_default is called back to update the maximum/minimum exposure time, maximum/minimum gain, maximum/minimum aperture, and AE compensation. Internally, AE updates u16HistRatioSlope and u8MaxHistOffset to their algorithm defaults. It is important to note that the exposure ratio Settings will not be updated and will remain the same as last time.
- au8AdjustTargetMin/au8AdjustTargetMax can brightness set different goals for different environment brightness min/big value, AE will be based on the current environment brightness convergence target brightness to the area in between
- u16LowBinThr/u16HighBinThr this parameter can be set to exclude AE metering, too dark and too bright part of the image, suggest u16LowBinThr not more than 10 (8 bits)

- enIRCutStatus s16IRCutOnLv/s16IRCutOffLv when product of the IR Cut for the swappable, this parameter can be used, when switching mode to Auto, depending On the setting of environmental brightness to switch On/Off the IR Cur. It is recommended that the brightness difference between the two environments should be more than 6 LV, so that frequent and repeated switching will not occur

【Related Data Type and Interface】

- *CVI_ISP_SetExposureAttr*

4.5.2.11 ISP_ME_ATTR_S

【Description】

Define manual exposure attributes.

【Syntax】

```
typedef struct _ISP_ME_ATTR_S

{

   ISP_OP_TYPE_E enExpTimeOpType;

   ISP_OP_TYPE_E enAGainOpType;

   ISP_OP_TYPE_E enDGainOpType;

   ISP_OP_TYPE_E enISPDGainOpType;

   CVI_U32 u32ExpTime;

   CVI_U32 u32AGain;

   CVI_U32 u32DGain;

   CVI_U32 u32ISPDGain;

   ISP_OP_TYPE_E enISONumOpType;

   ISP_AE_GAIN_TYPE_E enGainType;

   CVI_U32 u32ISONum;

} ISP_ME_ATTR_S;
```

【Member】

Member	Description
enExpTimeOpType	Manual exposure time enable, the default value is OP_TYPE_AUTO
enAGainOpType	Manual sensor analog gain enable, the default value is OP_TYPE_AUTO
enDGainOpType	Manual sensor digital gain enable, the default value is OP_TYPE_AUTO
enISPDGainOpType	Manual ISP digital gain enable, the default value is OP_TYPE_AUTO
u32ExpTime	Manual exposure time, in microseconds (us), the default value is 0x4000. Value range: [0x0, 0xFFFFFFFF], the specific range is related to the sensor.
u32AGain	Manual sensor analog gain, 10bit decimal precision, the default value is 0x400. Value range: [0x400, 0xFFFFFFFF], the specific range is related to the sensor.
u32DGain	Manual sensor digital gain, 10bit decimal precision, the default value is 0x400. Value range: [0x400, 0xFFFFFFFF], the specific range is related to the sensor.
u32ISPDGain	Manual ISP digital gain, 10bit decimal precision, the default value is 0x400. Value range: [0x400, 0x40000], the specific range is related to the sensor.
enISONumOpType	Choose to control the gain by ISO Num or gain, the default value is 0
enGainType	way of gain.
u32ISONum	Manual ISO Num gain, the default value is 100 , only works when enGainType = AE_TYPE_ISO.

【Note】

manual mode, manual exposure time and gain value will be automatically mode maximum/minimum exposure time and gain.

manual exposure can make effective parameters, you must set up corresponding manual exposure parameters, if not set, USES the system default.

gain unit for 10 bit decimal multiples of accuracy, namely 1024 representative 1 times.

If exposure parameter Settings beyond the maximum (small), will use the sensor to support maximum (small) instead.

【Related Data Type and Interface】

- *CVI_ISP_SetExposureAttr*

4.5.2.12 ISP_EXPOSURE_ATTR_S

【Description】

Define ISP exposure attributes.

【Syntax】

```
typedef struct _ISP_EXPOSURE_ATTR_S

{

   CVI_BOOL bByPass;

   ISP_OP_TYPE_E enOpType;

   CVI_U8 u8AERunInterval;

   CVI_BOOL bHistStatAdjust;

   CVI_BOOL bAERouteExValid;

   ISP_ME_ATTR_S stManual;

   ISP_AE_ATTR_S stAuto;

   CVI_U8 u8DebugMode;

   ISP_AE_METER_MODE_E enMeterMode;

   CVI_BOOL bAEGainSepCfg;

} ISP_EXPOSURE_ATTR_S;
```

【Member】

Member	Description
bByPass	The AE module bypass function is enabled, and the default is CVI_FALSE.
enOpType	Automatic exposure or manual exposure switch, the default is OP_TYPE_AUTO.
u8AERunInterval	The interval at which the AE algorithm runs, and the value range is [1, 255]. When the value is 1, it means that the AE algorithm is run every frame; When the value is 2, it means to run the AE algorithm every 2 frames, and so on. It is recommended not to set this value greater than 2, otherwise the AE adjustment speed will be affected. In WDR mode, it is recommended to set this value to 1, so that AE convergence will be smoother. This value defaults to 1.
bHistStatAdjust	This parameter will lower AE' s metering brightness for bright areas, making the bright areas brighter. For large-area sky scenes, if the metering is too dark, you can set this parameter Default is 0
bAERouteExValid	Whether the AE extended distribution route is valid or not. When CVI_TRUE, use the AE extended distribution route, otherwise use the normal AE distribution route. Defaults to CVI_FALSE.
stManual	Manual Exposure Properties Structure
stAuto	Auto Exposure Properties Structure
u8DebugMode	Set the debug mode of AE, used for debug functions such as dump log, normally 0
enMeterMode	AE metering method. 0 : AE_METER_MULTI central weighted metering. 1 : AE_METER_AVERAGE average weighted metering. 2 : AE_METER_HIGHLIHGT_PRIORITY Bright area priority metering, suitable for IR sensor use, the default value is AE_METER_MULTI
bAEGainSepCfg	Whether the long and short frame gains are allocated separately

【Note】

- When AE ByPass is CVI_TRUE, the AE module is bypassed, and any AE configuration will not affect the image brightness. ISP_AE_RESULT_S remains the value of the previous frame of AE bypass.
- When WDR/Linear mode is switched, u8AERunInterval will be reset to 1, and the user can modify this value as needed after the switch is completed.
- bAEGainSepCfg can only take effect when it is configured in 2to1WDR mode and the sensor supports this function.
- Manual mode does not support the separate distribution of long and short frame gains

【Related Data Type and Interface】

- *CVI_ISP_SetExposureAttr*
- *CVI_ISP_GetExposureAttr*

4.5.2.13 ISP_WDR_EXPOSURE_ATTR_S

【Description】

Define exposure attributes in WDR mode.

【Syntax】

```
#define EXP_RATIO_NUM (3)

typedef struct _ISP_WDR_EXPOSURE_ATTR_S

{

 ISP_OP_TYPE_E enExpRatioType;

 CVI_U32 au32ExpRatio[EXP_RATIO_NUM];

 CVI_U32 u32ExpRatioMax;

 CVI_U32 u32ExpRatioMin;

 CVI_U16 u16Tolerance;

 CVI_U16 u16Speed;

 CVI_U16 u16RatioBias;

 CVI_U8 u8SECompensation;

 CVI_U16 u16SEHisThr;

 CVI_U16 u16SEHisCntRatio1;

 CVI_U16 u16SEHisCntRatio2;

 CVI_U16 u16SEHis255CntThr1;

 CVI_U16 u16SEHis255CntThr2;

 CVI_U8 au8LEAdjustTargetMin[LV_TOTAL_NUM];

 CVI_U8 au8LEAdjustTargetMax[LV_TOTAL_NUM];

 CVI_U8 au8SEAdjustTargetMin[LV_TOTAL_NUM];
```

(continues on next page)

(continued from previous page)

```
    CVI_U8 au8SEAdjustTargetMax[LV_TOTAL_NUM];

    CVI_U8 u8AdjustTargetDetectFrmNum;

    CVI_U32 u32DiffPixelNum;

    CVI_U16 u16LELowBinThr;

    CVI_U16 u16LEHighBinThr;

    CVI_U16 u16SELowBinThr;

    CVI_U16 u16SEHighBinThr;

    CVI_U8 au8FrameAvgLumaMin[LV_TOTAL_NUM];

    CVI_U8 au8FrameAvgLumaMax[LV_TOTAL_NUM];

} ISP_WDR_EXPOSURE_ATTR_S;
```

【Member】

Member	Description
enExpRatioType	Only valid in multi-frame synthesis WDR mode. OP_TYPE_AUTO: Automatically calculate the exposure ratio of long and short frames according to the scene; OP_TYPE_MANUAL: Manually configure the exposure ratio of long and short frames.
au32ExpRatio	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, au32ExpRatio is invalid. When enExpRatioType is OP_TYPE_MANUAL, au32ExpRatio indicates the expected exposure ratio of two adjacent WDR frames for two-frame synthesis. Among them, au32ExpRatio[0] is the exposure ratio of long detection/short detection. 6bit decimal precision, 0x40 means the exposure ratio is 1 times. Value range: [0x40, 0xFFF]
u32ExpRatioMax	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMax indicates the maximum value of the ratio of the exposure time of the longest frame to the shortest frame. That is, when 2 frames are synthesized, it represents the maximum value of the long-detection/short-detection exposure ratio. u32ExpRatioMax is invalid when enExpRatioType is OP_TYPE_MANUAL. 6bit decimal precision, 0x40 means the exposure ratio is 1 times. Value range: [0x40, 0x4000]
u32ExpRatioMin	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMin indicates the minimum value of the ratio of long frame exposure time to short frame exposure time. When enExpRatioType is OP_TYPE_MANUAL, u32ExpRatioMin is invalid. The format is unsigned 6.6bit fixed point, 0x40 means that the ratio of long frame exposure time to short frame exposure time is 1 times. The default value is 0x40. Value range: [0x40, u32ExpRatioMax]
u16Tolerance	The tolerance deviation of the screen brightness during exposure adjustment is only valid in the two-frame composite WDR mode. The default is 5 Value range: [0x0, 0xFF]
u16Speed	The automatic exposure ratio adjustment speed is only valid in the two-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the larger the value, the faster the automatic exposure ratio adjustment. The default value is 0x20. Value range: [0x0, 0xFF]
u16RatioBias	The exposure ratio deviation value is only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the larger the value, the larger the automatic exposure ratio. The default value is 0x400, which means that the calculation result of the automatic exposure ratio algorithm will not be adjusted. The exposure ratio adjusted by this value will be limited by the maximum/minimum value of the exposure ratio. Value range: [0x0, 0xFFFF]
u8SECompensation	The target brightness of short detection, the default value is 56
au8LEAdjustTargetMin	Set the minimum value of the target brightness for each LV
	AE to converge to the long-term ambient brightness. Value range: (0x0 , 0x100), The default value of LV -5 ~ 15 is { 15, 15, 15, 15, 15, 15, 15, 15, 20, 20, 25, 30, 35, 40, 40, 50, 50, 55, 60,

【Note】

For some sensors that have a limit on the maximum exposure time of short frames, the exposure is relatively small, the maximum exposure time of long frames is short, and the image dynamic range and noise performance is poor, resulting in inaccurate automatic exposure ratio calculations. At this time, it is recommended to limit the minimum The exposure ratio ensures that the maximum exposure time of long frames reaches at least 3ms.

It is recommended not to set u16Speed to less than 0x8, to avoid the exposure ratio adjustment being too slow or even not adjusted due to insufficient calculation accuracy in some scenes. Excessive u16Speed may cause the exposure ratio to change too quickly, causing the brightness of the screen to oscillate.

It is recommended not to set u32ExpRatio greater than 0x400 in manual mode. If the exposure ratio is greater than 0x400, in a bright ultra-wide dynamic scene, appropriately increasing the exposure ratio will optimize the long- frame image noise performance. However, in dark or low wide dynamic scenes, if the exposure ratio is too large, the maximum exposure time of short frames will be compressed, resulting in poor image noise performance, and there will be obvious noise discontinuity, and motion performance will also deteriorate.

【Related Data Type and Interface】

- *CVI_ISP_SetWDRExposureAttr*
- *CVI_ISP_GetWDRExposureAttr*

4.5.2.14 ISP_AE_ROUTE_NODE_S

【Description】

Define AE assignment route node attributes.

【Syntax】

```
typedef struct _ISP_AE_ROUTE_NODE_S

{

 CVI_U32 u32IntTime;

 CVI_U32 u32SysGain;

 ISP_IRIS_F_NO_E enIrisFNO;

 CVI_U32 u32IrisFNOLin;

} ISP_AE_ROUTE_NODE_S;
```

【Member】

Member	Description
u32IntTime	Node exposure time, in microseconds (us).Value range: (0x0, 0xFFFFFFFF]
u32SysGain	Node gain, including sensor analog gain, sensor digital gain and ISP digital gain, 10bit precision.Value range: [0x400, 0xFFFFFFFF]
enIrisFNO	Node aperture F value size, only supports P-Iris, not DC-Iris.Value range: [ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0].
u32IrisFNOLin	Node aperture F value equivalent gain size, only supports P-Iris, not DC-Iris.Value range: [1, 1024]

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetAERouteAttr*

4.5.2.15 ISP_AE_ROUTE_S

【Description】

Define AE exposure assignment policy attributes.

【Syntax】

```
#define ISP_AE_ROUTE_MAX_NODES (16)

typedef struct _ISP_AE_ROUTE_S

{

CVI_U32 u32TotalNum;

ISP_AE_ROUTE_NODE_S astRouteNode[ISP_AE_ROUTE_MAX_NODES];

} ISP_AE_ROUTE_S;
```

【Member】

Member	Description
u32TotalNum	The number of exposure distribution route nodes, currently the maximum is 16.
astRouteNode [ISP_AE_ROUTE_MAX_NODES]	Exposure assignment route node properties.

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetAERouteAttr*

4.5.2.16 ISP_AE_ROUTE_EX_NODE_S

【Description】

Define the attribute of AE extension assignment route node.

【Syntax】

```
typedef struct _ISP_AE_ROUTE_EX_NODE_S

{

 CVI_U32 u32IntTime;

 CVI_U32 u32Again;

 CVI_U32 u32Dgain;

 CVI_U32 u32IspDgain;

 ISP_IRIS_F_NO_E enIrisFNO;

 CVI_U32 u32IrisFNOLin;

} ISP_AE_ROUTE_EX_NODE_S;
```

【Member】

Member	Description
u32IntTime	Node exposure time, in microseconds (us).Value range: (0x0, 0xFFFFFFFF]
u32Again	Sensor analog gain, 10bit precision.Value range: [0x400, 0x3FFFFF]
u32Dgain	Sensor digital gain, 10bit precision.Value range: [0x400, 0x3FFFFF]
u32IspDgain	ISP digital gain, 10bit precision.Value range: [0x400, 0x40000]
enIrisFNO	Node aperture F value size, only supports P-Iris, not DC-Iris.Value range: [ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]
u32IrisFNOLin	Node aperture F value equivalent gain size, only supports P-Iris, not DC-Iris.Value range: [1, 1024]

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetAERouteAttrEx*

4.5.2.17 ISP_AE_ROUTE_EX_S

【Description】

Define extended attributes of AE exposure allocation strategy.

【Syntax】

```
#define ISP_AE_ROUTE_EX_MAX_NODES (16)

typedef struct _ISP_AE_ROUTE_EX_S

{

CVI_U32 u32TotalNum;

ISP_AE_ROUTE_EX_NODE_S astRouteExNode[ISP_AE_ROUTE_EX_MAX_NODES];

} ISP_AE_ROUTE_EX_S;
```

【Member】

Member	Descirption
u32TotalNum	The number of exposure extension distribution route nodes is currently up to 16.
astRouteExNode [ISP_AE_ROUTE_EX_MAX_NODES]	The exposure extension assigns route node properties.

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetAERouteAttrEx*

4.5.2.18 ISP_EXP_INFO_S

【Description】

Define the internal state information of ISP exposure.

【Syntax】

```
#define HIST_NUM (256)

 typedef struct _ISP_EXP_INFO_S

 {

  CVI_U32 u32ExpTime;

  CVI_U32 u32ShortExpTime;
```

(continues on next page)

(continued from previous page)

```
CVI_U32 u32MedianExpTime;
CVI_U32 u32LongExpTime;
CVI_U32 u32AGain;
CVI_U32 u32DGain;
CVI_U32 u32ISPDGain;
CVI_U32 u32Exposure;
CVI_BOOL bExposureIsMAX;
CVI_S16 s16HistError;
CVI_U32 au32AE_Hist256Value[HIST_NUM];
CVI_U8 u8AveLum;
CVI_U32 u32LinesPer500ms;
CVI_U32 u32PirisFNO;
CVI_U32 u32Fps;
CVI_U32 u32ISO;
CVI_U32 u32ISOCalibrate;
CVI_U32 u32RefExpRatio;
CVI_U32 u32FirstStableTime;
ISP_AE_ROUTE_S stAERoute;
ISP_AE_ROUTE_EX_S stAERouteEx;
CVI_U8 u8WDRShortAveLuma;
CVI_U32 u32WDRExpRatio;
CVI_U8 u8LEFrameAvgLuma;
CVI_U8 u8SEFrameAvgLuma;
CVI_FLOAT fLightValue;
```

(continues on next page)

(continued from previous page)

```
    CVI_U32 u32AGainSF;
    CVI_U32 u32DGainSF;
    CVI_U32 u32ISPDGainSF;
    CVI_U32 u32ISOSF;
    ISP_AE_ROUTE_S stAERouteSF;
    ISP_AE_ROUTE_EX_S stAERouteSFEx;
    CVI_BOOL bGainSepStatus;
}ISP_EXP_INFO_S;
```

【Member】

Member	Description
u32ExpTime	The current exposure time, in microseconds (us). Value range: [0x0, 0xFFFFFFFF]
u32ShortExpTime	In FSWDR mode, it indicates the exposure time of the current short frame (S), in microseconds (us). Linear mode does not care about this value. Value range: [0x0, 0xFFFFFFFF]
u32LongExpTime	In FSWDR mode, it indicates the current long frame exposure time in microseconds (us). Value range: [0x0, 0xFFFFFFFF]
u32AGain	Current sensor analog gain, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF]
u32DGain	Current sensor digital gain, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF]
u32ISPDGain	Current ISP digital gain, 10bit decimal precision. Value range: [0x400, 0xFFFFFFFF]
u32Exposure	The current exposure is equal to the product of exposure time and exposure gain, where the unit of exposure time is the number of exposure lines, and the exposure gain is 6bit decimal precision. Value range: [0x40, 0xFFFFFFFF]
bExposureIsMAX	0: ISP has not reached the maximum exposure level; 1: ISP reaches maximum exposure level.
s16HistError	Statistical information, the difference between the target luminance value of AE and the actual value. A positive value indicates that the current expected luminance information is greater than the actual luminance information. A negative value indicates that the expected luminance information is smaller than the actual luminance information.
au32AE_Hist256Value	Global 256-segment histogram statistics Value range: [0x0, 0xFFFFFFFF]
u8AveLum	average brightness information Value range: [0x0, 0xFF]
u32LinesPer500ms	The current number of exposure lines corresponding to every 500ms can be used to convert the unit of exposure time from us to the number of lines. Value range: [0x0, 0xFFFFFFFF]
u32PirisFNO	The equivalent gain corresponding to the current P-Iris aperture F value. Value range: [0x0, 0x400]
u32Fps	Actual image frame rate * 100. Value range: [0x0, 0xFFFFFFFF]
u32ISO	Current sensor analog gain * sensor digital gain * ISP digital gain * 100, where the accuracy of the gain is 10bit. Value range: [0x64, 0xFFFFFFFF]
u32ISOCalibrate	Standard ISO, used for displaying DCF information in photos. u32ISOCalibrate = u32ISO * 256 / u16ISOCalCoef.
u32RefExpRatio	The reference exposure ratio, used to estimate the dynamic range of the current scene, will be affected by the values of Tolerance and Speed in ISP_WDR_EXPOSURE_ATTR_S. Value range: [0x40, 0x4000]
u32FirstStableTime	The time for the first AE to converge and stabilize, in microseconds (us). Value range: [0x0, 0xFFFFFFFF]
stAERoute	For the actual effective AE route, the exposure time in each node is in us, the gain is 10bit precision, and the aperture value range is [ISP_IRIS_F_NO_32_0, ISP_IRIS_F_NO_1_0]. When the aperture type is DC-Iris, the node aperture value will not affect the exposure distribution.
stAERouteEx	For the extended AE route that actually takes effect, the ex-

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_QueryExposureInfo*

4.5.3 IRIS

- *ISP_IRIS_STATUS_E* :Define ISP aperture status.
- *ISP_IRIS_TYPE_E* :Define ISP aperture type.
- *ISP_IRIS_F_NO_E* :Define ISP aperture F value.
- *ISP_MI_ATTR_S* :Define manual aperture attributes.
- *ISP_DCIRIS_ATTR_S* :Define DC-Iris Deep Learning algorithm attributes.
- *ISP_PIRIS_ATTR_S* :Define the P-Iris attribute.
- *ISP_IRIS_ATTR_S* :Define the ISP aperture properties.

4.5.3.1 ISP_IRIS_STATUS_E

【Description】

Define ISP aperture status.

【Syntax】

```
typedef enum _ISP_IRIS_STATUS_E

{

ISP_IRIS_KEEP = 0,

ISP_IRIS_OPEN = 1,

ISP_IRIS_CLOSE = 2,

ISP_IRIS_BUTT

} ISP_IRIS_STATUS_E;
```

【Member】

Member	Description
ISP_IRIS_KEEP	The aperture maintains its current state.
ISP_IRIS_OPEN	Aperture wide open.
ISP_IRIS_CLOSE	Aperture fully closed.

【Note】

When the value is set to ISP_IRIS_OPEN or ISP_IRIS_CLOSE, the iris is fully open or fully closed, which can be used to test whether the Deep Learning circuit and driver are correct.

The priority of OPEN and CLOSE is higher than that of Deep Learning enable and manual/auto mode.

When the Deep Learning algorithm is running, in order to ensure its normal operation, the value needs to be set to ISP_IRIS_KEEP.

【Related Data Type and Interface】

None.

4.5.3.2 ISP_IRIS_TYPE_E

【Description】

Define ISP aperture type.

【Syntax】

```
typedef enum _ISP_IRIS_TYPE_E

{

ISP_IRIS_DC_TYPE = 0,

ISP_IRIS_P_TYPE,

ISP_IRIS_TYPE_BUTT,

} ISP_IRIS_TYPE_E;
```

【Member】

Member	Description
ISP_IRIS_DC_TYPE	DC-Iris aperture
ISP_IRIS_P_TYPE	P-Iris aperture

【Note】

The correct aperture type must be set for the Deep Learning algorithm to work properly.

If it is connected with a manual iris lens, this value can be set to ISP_IRIS_DC_TYPE, and it is recommended to disable Deep Learning enablement at this time.

【Related Data Type and Interface】

None.

4.5.3.3 ISP_IRIS_F_NO_E

【Description】

Define ISP aperture F value.

【Syntax】

```
typedef enum _ISP_IRIS_F_NO_E

{

 ISP_IRIS_F_NO_32_0 = 0,ISP_IRIS_F_NO_22_0,ISP_IRIS_F_NO_16_0,ISP_IRIS_F_NO_11_
↪0,ISP_IRIS_F_NO_8_0,ISP_IRIS_F_NO_5_6,ISP_IRIS_F_NO_4_0,ISP_IRIS_F_NO_2_
 8,ISP_IRIS_F_NO_2_0,ISP_IRIS_F_NO_1_4,ISP_IRIS_F_NO_1_0,ISP_IRIS_F_NO_BUTT,

} ISP_IRIS_F_NO_E;
```

【Member】

Member	Description	Equivalent Gain
ISP_IRIS_F_NO_32_0	Aperture F32.0	1
ISP_IRIS_F_NO_22_0	Aperture F22.0	2
ISP_IRIS_F_NO_16_0	Aperture F16.0	4
ISP_IRIS_F_NO_11_0	Aperture F11.0	8
ISP_IRIS_F_NO_8_0	Aperture F8.0	16
ISP_IRIS_F_NO_5_6	Aperture F5.6	32
ISP_IRIS_F_NO_4_0	Aperture F4.0	64
ISP_IRIS_F_NO_2_8	Aperture F2.8	128
ISP_IRIS_F_NO_2_0	Aperture F2.0	256
ISP_IRIS_F_NO_1_4	Aperture F1.4	512
ISP_IRIS_F_NO_1_0	Aperture F1.0	1024

【Note】

For P-Iris, when the AE algorithm calculates the exposure according to the distribution route, the aperture F value should be equivalent to a gain, the formula is as follows: equivalent gain FNO = 1 << ISP_IRIS_F_NO_XX_XX.

It can be seen that F32.0 corresponds to gain 1, F22.0 corresponds to gain 2, F16.0 corresponds to gain 4, and so on, F1.0 corresponds to gain 1024.

【Related Data Type and Interface】

None.

4.5.3.4 ISP_MI_ATTR_S

【Description】

Define manual aperture attributes.

【Syntax】

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

【Member】

Member	Description
u32HoldValue	Deep Learning correction value, used for DC-Iris debugging.The value range is [0x0, 0x3E8].
enIrisFNO	The size of the manual aperture is distinguished according to the aperture F value, only supports P-Iris, not DC-Iris.

【Note】

When docking DC-Iris lens, if ISP_IRIS_STATUS_E is set to ISP_IRIS_KEEP, manual iris is enabled, u32HoldValue can be used for DC-Iris debugging, and the duty cycle of PWM at this time is u32HoldValue.

When docking a P-Iris lens, if ISP_IRIS_STATUS_E is set to ISP_IRIS_KEEP, the manual iris is enabled, and enIrisFNO can be used for P-Iris debugging. At this time, the P-Iris stepping motor will be controlled to move to the position where the iris F value is closest to enIrisFNO. In the automatic exposure mode, the P-Iris manual aperture does not take effect. If you want to fix the aperture to a certain F value, you can set enMaxIrisFNOTarget/ enMinIrisFNOTarget to the same value to achieve.

【Related Data Type and Interface】

None.

4.5.3.5 ISP_DCIRIS_ATTR_S

【Description】

Define DC-Iris Deep Learning algorithm attributes.

【Syntax】

```
typedef struct _ISP_DCIRIS_ATTR_S

{

 CVI_S32 s32Kp;

```

(continues on next page)

(continued from previous page)

```
 CVI_S32 s32Ki;

  CVI_S32 s32Kd;CVI_U32 u32MinPwmDuty;CVI_U32 u32MaxPwmDuty;CVI_U32␣
↪u32OpenPwmDuty;

} ISP_DCIRIS_ATTR_S;
```

【Member】

Member	Description
s32Kp	Proportional gain, used to adjust the opening and closing speed of the aperture, the larger the value is, the faster the opening and closing speed of the aperture will be. If the value is too small, it is easy to cause oscillation during the convergence process of light changes, and if the value is too large, it is easy to overshoot and cause oscillation. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 7000. The value range is [0, 100000].
s32Ki	Integral gain, used to adjust the opening and closing speed of the aperture, the larger the value, the faster the opening and closing speed of the aperture. When the value is small, the screen will stabilize at a relatively low brightness when the strong light converges; when the value is large, the aperture may not be closed in strong light scenes. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 100, which generally does not need to be modified. The value range is [0, 1000].
s32Kd	Differential gain, used to limit the opening and closing speed of the aperture when the light changes drastically, the larger the value is, the opening and closing speed of the aperture when the light changes drastically slower. If the value is too large, it is too sensitive to instantaneous brightness changes, which will cause the screen to oscillate when the scene brightness changes rapidly. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 3000. The value range is [0, 100000].
u32MinPwmDuty	Minimum PWM duty cycle. The smaller the value, the faster the closing speed of the aperture will be when the overexposure is over, but it will easily cause the aperture to oscillate back and forth. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 250. The value range is [0, 1000].
u32MaxPwmDuty	Maximum PWM duty cycle. The larger the value is, the faster the aperture will open when the screen is completely dark. If the value is too small, the aperture may not reach the maximum when exiting the aperture control area, resulting in serious image noise. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 950. The value range is [0, 1000].
u32OpenPwmDuty	PWM duty cycle when the iris is open. When the screen brightness is stable and the PWM duty cycle is greater than this value for a period of time, exit the aperture control area area. Therefore, the value should not be too small, otherwise it will easily cause the aperture to exit the aperture control area before reaching the maximum, resulting in serious image noise. The reasonable setting of this value is related to circuit characteristics and lens. The recommended value is 800. The value range is [0, 1000].

【Note】

When the iris closes and oscillates, it generally means that the iris is closed too fast, which can be solved by appropriately reducing s32Kp and increasing u32MinPwmDuty.

The value of u32OpenPwmDuty is required to be between u32MinPwmDuty and u32MaxPwmDuty, and it must be ensured that this value can open the aperture faster.

【Related Data Type and Interface】

- *CVI_ISP_SetDcirisAttr*
- *CVI_ISP_GetDcirisAttr*

4.5.3.6 ISP_PIRIS_ATTR_S

【Description】

Define the P-Iris attribute.

【Syntax】

None.

【Member】

【Note】

Support for P-Iris is currently not implemented

【Related Data Type and Interface】

None.

4.5.3.7 ISP_IRIS_ATTR_S

【Description】

Define the ISP aperture properties.

【Syntax】

```
typedef struct _ISP_IRIS_ATTR_S {

 CVI_BOOL bEnable;

ISP_OP_TYPE_E enOpType;

ISP_IRIS_TYPE_E enIrisType;

ISP_IRIS_STATUS_E enIrisStatus;

ISP_MI_ATTR_S stMIAttr;

} ISP_IRIS_ATTR_S;
```

【Member】

Member	Description
bEnable	Auto iris enabled.
enOpType	Auto iris or manual iris mode selection.
enIrisType	Aperture type, DC-Iris or P-Iris.
enIrisStatus	Aperture status.
stMIAttr	Manual aperture property setting structure.

【Note】

When the value is set to ISP_IRIS_OPEN or ISP_IRIS_CLOSE, the iris is fully open or fully closed, which can be used to test whether the Deep Learning circuit and driver are correct. The priority of OPEN and CLOSE is higher than that of Deep Learning enable and manual/auto mode. When the Deep Learning algorithm is running, in order to ensure its normal operation, the value needs to be set to ISP_IRIS_KEEP.

【Related Data Type and Interface】

- *CVI_ISP_SetIrisAttr*
- *CVI_ISP_GetIrisAttr*

5 AWB

5.1 Overview

The color temperature changes with the spectral composition of visible light. In low color temperature light source, white objects tend to be red. In high color temperature light source, white objects tend to be blue. The human eye can judge the real color of objects according to the memory of the brain. The function of AWB is to reduce the influence of external light sources on the real color of objects, so that the color information we collect can be transformed into unbiased color information under the ideal sunlight source.

5.2 Important Concepts

- Color temperature: the color temperature is defined by the absolute blackbody. When the radiation of the light source in the visible area is exactly the same as that of the absolute blackbody, the temperature of the blackbody is called the color temperature of the light source.
- White balance: under the light source of different color temperature, the response of white in the sensor will be blue or red. White balance algorithm adjusts the intensity of R, G, B three color channels to make white real.

5.3 Function Overview

AWB module consists of WB information statistics module and AWB strategy control algorithm firewall. The WB information statistics module of ISP determines whether each pixel output by sensor meets the white dot condition set by the user, and calculates the average value of R, G and B color channels of all pixels meeting the condition.

The image can be divided into m * n (M rows and N columns) regions, and the R, G and B mean values of each region and the number of white points participating in the statistics can be counted.

Support the output of R, G, B means of the whole image and the number of white points participating in the statistics.

The working principle of AWB is shown in Figure 4-1.

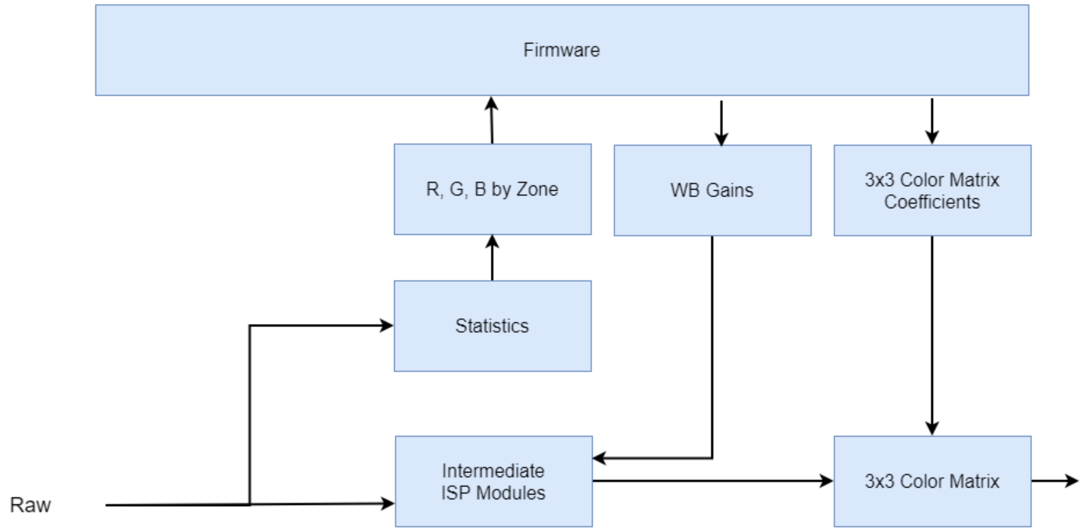


Fig. 5.1: Figure 4-1 AWB working principle

5.4 API Reference

5.4.1 AWB library interface

All AWB library interfaces are only for CVI AWB library. If customers implement AWB library by themselves, they don’ t need to care for these interfaces and can’ t use them.

- *CVI_AWB_Register* : Register AWB library with ISP.
- *CVI_AWB_UnRegister* : Inject AWB library to ISP.
- *CVI_AWB_SensorRegCallBack* : The callback interface of sensor registration provided by AWB library.
- *CVI_AWB_SensorUnRegCallBack* : The callback interface of sensor annotation provided by AWB library.

5.4.1.1 CVI_AWB_Register

【Description】

Register AWB library with ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAwbLib	AWB algorithm library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files：cvi_awb.h
- Library files：libawb.a

【Note】

- **This interface calls `CVI_ISP_AWBLibRegCallBack` , which is the callback interface of AWB registration p** ISP library, to realize the function of AWB registering with ISP library.
- This interface does not support multi process operation.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None

【Related Topic】

None

5.4.1.2 CVI_AWB_UnRegister

【Description】

Inject AWB library to ISP.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAwbLib	AWB algorithm library structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- **The interface calls the callback interface CVI_ISP_AWBLibUnRegCallBack of AWB anti registration prov** ISP library to realize the function of AWB anti registration with ISP library.
- This interface does not support multi-process operations.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None

【Related Topic】

None

5.4.1.3 CVI_AWB_SensorRegCallBack

【Description】

The callback interface of sensor registration provided by AWB library.

【Syntax】

```
CVI_S32 CVI_AWB_SensorRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib, ISP_SNS_ATTR_INFO_S *pstSnsAttrInfo, AWB_SENSOR_REGISTER_S *pstRegister);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAeLib	AWB algorithm library structure pointer	Input
pstSnsAttrInfo	Properties of sensors registered with AWB	Input
pstRegister	Sensor register structure pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- SensorId is a custom value in the sensor library. It is mainly used to check whether the sensor registered with ISP and 3A are the same sensor.

- Through a series of callback interfaces registered by sensor, AWB obtains differentiated initialization parameters and controls sensor.
- This interface does not support multi process operation.
- This interface is not supported on the linux side of the dual-os SDK.

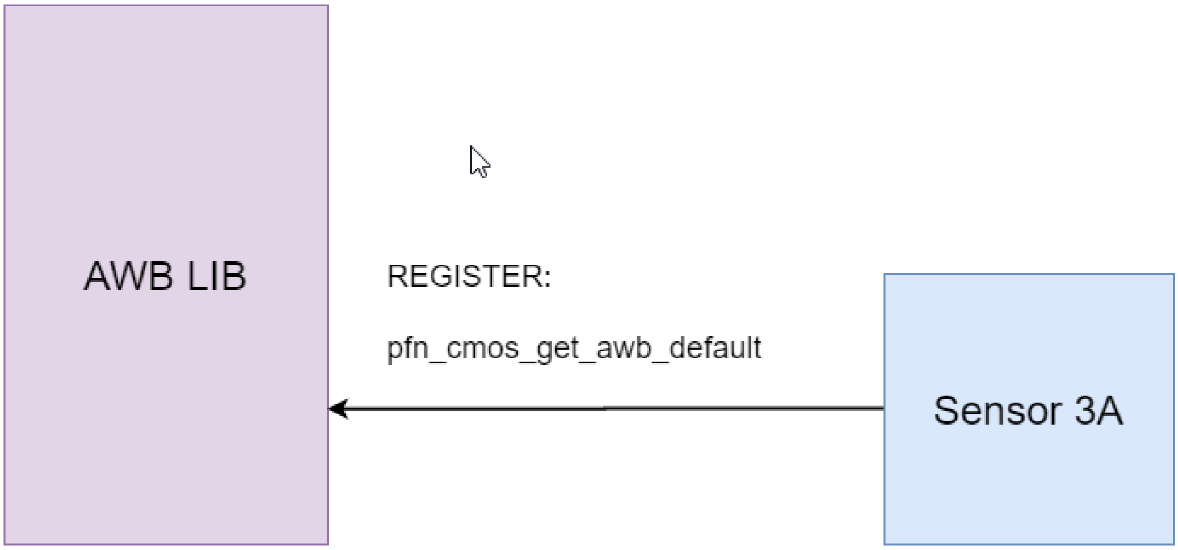


Fig. 5.2: Figure 4-2 interface between AWB library and sensor Library

【Example】

None

【Related Topic】

None

5.4.1.4 CVI_AWB_SensorUnRegCallBack

【Description】

The callback interface for sensor de-registration provided by the AWB library.

【Syntax】

```
CVI_S32 CVI_AWB_SensorUnRegCallBack(VI_PIPE ViPipe, ALG_LIB_S *pstAwbLib,
SENSOR_ID SensorId);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAwbLib	AWB algorithm library structure pointer	Input
SensorId	Id of sensor de-registered with AWB	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- **SensorId is a custom value in the sensor library. It is mainly** used to check whether the sensor registered with ISP and 3A are the same sensor.
- This interface does not support multi process operation.
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None

【Related Topic】

None

5.4.2 AWB Control Module

- *CVI_ISP_SetWBAttr* : Set white balance attributes.
- *CVI_ISP_GetWBAttr* : Get the white balance attribute.
- CVI_ISP_SetWBAttrEx :set the white balance extension property.
- CVI_ISP_GetWBAttrEx :get the white balance extension property.
- *CVI_ISP_QueryWBINfo* : Get the current white balance gain coefficient, detect the color temperature, saturation value,color correction matrix coefficient.
- *CVI_ISP_SetAWBLogPath* :When using CVI Awb lib, save the path of AWB debug log.
- *CVI_ISP_SetAWBLogName* :When using CVI Awb lib, store the name of AWB debug log.
- *CVI_ISP_GetGrayWorldAwbInfo* :Get gray world WB information.

5.4.2.1 CVI_ISP_SetWBAttr

【Description】

Set white balance attributes.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstWBAttr	White balance attribute struct pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- When the white balance control type is automatic, AWB algorithm automatically adjusts the white balance coefficient.
- When the white balance control type is manual, the AWB algorithm is invalid, and user needs to set Rgain, Ggain, Bgain on one’s own.
- The environment color temperature and illumination will affect the distribution of white spots. During the operation of CVI AWB algorithm, the white spot parameters of AWB Byaer statistical information will be automatically refreshed according to the environment parameters. If users want to modify AWB statistical parameters when CVI AWB algorithm is running, they need to call CVI_ISP_SetWBAttr interface to turn off the automatic refresh function of statistical parameters. When the user turns off the automatic refresh function of statistical parameters, there is a delay of 2 frames when the ISP receives the configuration and response. Therefore, the user needs to wait 2 frames before modifying the AWB statistical parameters.

【Example】

None

【Related Topic】

- *CVI_ISP_GetWBAttr*

5.4.2.2 CVI_ISP_GetWBAttr

【Description】

Get the white balance attributes.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstWBAttr	White balance attribute struct pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_SetWBAttr*

5.4.2.3 CVI_ISP_SetAWBAttrEx

【Description】

Set the white balance extension attributes.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
PstAWBAttrEx	Extended white balance attribute struct pointer	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

This interface is valid only when the CVI_ISP_SetWBAttr interface member pstWBAttr->enAlgType is AWB_ALG_ADVANCE.

【Example】

None

【Related Topic】

- *CVI_ISP_GetAWBAttrEx*

5.4.2.4 CVI_ISP_GetAWBAttrEx

【Description】

Get the white balance extension attributes.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
pstAWBAttrEx	Extended white balance attribute struct pointer	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_SetAWBAttrEx*

5.4.2.5 CVI_ISP_QueryWBInfo

【Description】

Get the current white balance gain coefficient, detect the color temperature, saturation value, color correction matrix coefficient.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe number	Input
PstWBInfo	Color related state parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

None

【Example】

None

【Related Topic】

None

5.4.2.6 CVI_ISP_SetAWBLogPath

【Description】

When using CVI Awb lib, save the path of AWB debug log.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
szPath	debug file path	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- The default path is /var/log
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None

【Related Topic】

None

5.4.2.7 CVI_ISP_SetAWBLogName

【Description】

When using CVI Awb lib, store the name of AWB debug log.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
szName	File name	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_awb.h
- Library files: libawb.a

【Note】

- The preset file name is AwbLog0.txt
- This interface is not supported on the linux side of the dual-os SDK.

【Example】

None

【Related Topic】

None

5.4.2.8 CVI_ISP_GetGrayWorldAwbInfo

【Description】

Get gray world WB information.

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	ViPipe id	Input
pRgain	R channel gain	Output
pBgain	B channel gain	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files：cvi_awb.h
- Library files：libawb.a

【Note】

None

【Example】

None

【Related Topic】

None

5.5 Date Types

5.5.1 Register

- AWB_SENSOR_REGISTER_S :Define the sensor registration structure.
- AWB_SENSOR_EXP_FUNC_S :Define the sensor callback function structure.
- AWB_SENSOR_DEFAULT_S :Define the initialization parameter structure of AWB algorithm library.
- AWB_SPEC_SENSOR_DEFAULT_S :Define the initialization parameter structure of SPECAWB algorithm.
- AWB_AGC_TABLE_S : Define the saturation initialization parameter structure.
- AWB_CCM_TAB_S : The automatic color correction matrix coefficients at different color temperatures are defined.
- AWB_CCM_S : Define CCM color correction matrix attributes.

5.5.2 WB

- *ISP_AWB_CBCR_TRACK_ATTR_S* :Define the linkage parameters of Bayer domain statistics
- *ISP_AWB_LUM_HISTGRAM_ATTR_S* :The brightness histogram statistical parameters of white balance are defined
- *ISP_AWB_ALG_E* :Define the calculation method property of white balance.
- *ISP_AWB_CT_LIMIT_ATTR_S* :Define the gain range limit attribute of white balance.
- *ISP_MWB_ATTR_S* :Define ISP manual white balance properties.
- *ISP_WB_ATTR_S* :Define the white balance attribute.
- *ISP_AWB_ATTR_S*: Define ISP automatic white balance properties.
- *ISP_AWB_ALG_TYPE_E*: Define AWB algorithm properties.
- *ISP_AWB_ATTR_EX_S* : Define the auto white balance extension attribute
- *ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S* :Define information about individual light points
- *ISP_AWB_IN_OUT_ATTR_S* :Define information about individual light points
- *ISP_AWB_MULTI_LS_TYPE_E* :Define AWB strategy under mixed light source
- *ISP_AWB_INDOOR_OUTDOOR_STATUS_E* : Define AWB indoor and outdoor state.
- *ISP_WB_INFO_S* :Define white balance, saturation, color correction information.

5.5.2.1 ISP_MWB_ATTR_S

【Description】

Define ISP manual white balance properties.

【Syntax】

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

【Member】

Member	Description
u16Rgain	R channel gain at MWB
u16Grgain	Gr channel gain at MWB
u16Grgain	Gb channel gain at MWB
u16Bgain	B channel gain at MWB

【Note】

A doubled gain of RGB channel is 0x400

【Related Data Type and Interface】

None

5.5.2.2 ISP_AWB_CT_LIMIT_ATTR_S

【Description】

Define the gain range limit attribute of white balance.

【Syntax】

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

【Member】

Member	Description
bEnable	awb gain range limiting switch
enOpType	Automatically or manually set the gain range of self balancing
u16HighRgLimit	Maximum R gain at high color temperature in manual mode
u16HighBgLimit	Minimum B gain at high color temperature in manual mode
u16LowRgLimit	Minimum R gain at low color temperature in manual mode
u16LowBgLimit	Maximum B gain at low color temperature in manual mode

【Note】

In automatic mode, AWB algorithm will calculate the upper and lower limits of gain by itself. When switching to manual mode, the above four R.B. gain limits will be used.

One time gain is 0x400

【Related Data Type and Interface】

None

5.5.3 ISP_AWB_ALG_E

【Description】

Define the calculation method property of white balance.

【Syntax】

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

【Member】

Member	Description
AWB	General AWB algorithm
ALG_AWB_SPEC	AWB algorithm based on machine learning is not supported at present.

【Note】

None

【Related Data Type and Interface】

None

5.5.3.1 ISP_AWB_ATTR_S

【Description】

Define automatic white balance properties.

【Syntax】

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

【Member】

Member	Description
bEnable	Automatic white balance algorithm switch
u16RefColorTemp	The color temperature value of static white balance light source is recommended as 5000K light source
au16StaticWB	Static white balance coefficient is given by AWB calibration tool. Value range: [0-0x3FFF]。
as32CurvePara	CurvePara[0-2] Planck curve coefficient is given by AWB calibration tool. Planck curve describes the color performance of white block under standard light source with different color temperature. CurvePara[3-5] color temperature curve coefficient, given by AWB calibration tool. The color temperature curve describes the corresponding relationship between the color performance of white block and color temperature.
enAlgType	AWB algorithm categories selection includes AWB_ALG_LOWCOST and AWB_ALG_ADVANCE. ISP_AWB_ATTR_EX_S extended attributes only work in AWB_ALG_ADVANCE.
u8RGStrength	Automatic white balance R channel calibration intensity. Value range: [0, 255]
u8BGStrength	Automatic white balance B channel calibration intensity. Value range: [0, 255]
u16Speed	The convergence speed of the automatic white balance algorithm is improved. Value range: [0, 4095]
u16ZoneSel	When the parameter is 0 or 255, the white balance algorithm similar to gray world is adopted, and other values are classified and screened to improve the accuracy
u16HighColorTemp	The upper limit of color temperature of automatic white balance algorithm. Value range: [8000, 10000]
u16LowColorTemp	The lower limit of color temperature for automatic white balance algorithm. Value range: [0x0, u8HighColorTemp)
stCTLimit	Manually or automatically limit the white balance gain value
bShiftLimitEn	AWB gain beyond white point range maps back to white point range switch
u16ShiftLimit	AWB calculation of white point range parameters
bGainNormEn	By limiting the gain of RGB channel, the signal-to-noise ratio of low color temperature and low illumination scenes can be improved
bNaturalCastEn	AWB style switch at low color temperature. Default to off.
stCbCrTrack	AWB statistical range and ISO linked parameters
stLumaHist	AWB brightness and weight parameters
bAWBZoneWtEn	Screen partition weight switch, preset as off.
au8ZoneWt	32x32 image weight, value range [0,0xff] (not implemented yet)

【Note】

None

【Related Data Type and Interface】

None

5.5.3.2 ISP_AWB_ALG_TYPE_E

【Description】

Define AWB algorithm properties.

【Syntax】

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

【Member】

Member	Description
AWB_ALG_LOWCOST	A simple and low computation AWB algorithm
AWB_ALG_ADVANCE	Advanced extended AWB algorithm, related to ISP_AWB_ATTR_EX_S
AWB_ALG_BUTT	Invalid setting

【Note】

None

【Related Data Type and Interface】

- *ISP_AWB_ATTR_EX_S_‘*

5.5.3.3 ISP_AWB_CBCR_TRACK_ATTR_S

【Description】

Define AWB statistical range and ISO linked parameters

【Syntax】

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

【Member】

Member	Description
bEnable	AWB statistical range and ISO linked switch
au16CrMax	The maximum value of R/G under different ISO
au16CrMin	The minimum value of R/G under different ISO
au16CbMax	The maximum value of B/G under different ISO
au16CbMin	The minimum value of B/G under different ISO

【Note】

It is recommended to calibrate CrMax(R/G) ,CbMin(B/G) at low color temperature.

【Related Data Type and Interface】

None

5.5.3.4 ISP_AWB_LUM_HISTGRAM_ATTR_S

【Description】

Define AWB brightness and weight parameters

【Syntax】

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

Macro definition is as follow

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

【Member】

Member	Description
bEnable	Whether or not to turn on the weight for different brightness, preset as open
enOpType	Automatic mode: AWB assigns weight automatically. Manual mode: users can set brightness classification and weight by themselves
au8HistThresh	Threshold of brightness classification (valid in manual mode)
au16HistWt	Weight of brightness classification (effective in manual mode)

【Note】

au8HistThresh[0] is fixed to 0, au8HistThresh[5] is fixed to 255.

au8HistThresh[i+1] must be greater than au8HistThresh[i]

au16HistWt weight value range is 32~512

【Related Data Type and Interface】

None

5.5.4 ISP_WB_ATTR_S

【Description】

Define white balance attributes

【Syntax】

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

【Member】

Member	Description
bByPass	White balance module Bypass enable, default to CVI_FALSE.
u8AWBRunInterval	Working frequency of white balance module. Value range: [0x1, 0xFF]
enOpType	Automatic white balance and manual white balance switch.
stManual	Manual parameters
stAuto	Automatic parameters
enAlgType	AWB algorithm type
u8DebugMode	It is used in debugging, and generally does not need to be set

【Note】

When bByPass is TRUE, other WB parameters will not take effect, and the RGB channel gain is fixed to double 0x400

The default value of u8AWBRunInterval is 6, which means AWB is executed every 6 frames. This value can be increased according to the demand to reduce the frequency of AWB.

【Related Data Type and Interface】

None

5.5.4.1 ISP_AWB_ATTR_EX_S

【Description】

Define auto white balance extended attributes

【Syntax】

```
typedef struct _ISP_AWB_ATTR_EX_S {
  CVI_U8 u8Tolerance;
  CVI_U8 u8ZoneRadius;
  CVI_U16 u16CurveLLimit;
  CVI_U16 u16CurveRLimit;
  CVI_BOOL bExtraLightEn;
  ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S stLightInfo[AWB_LS_NUM];
  ISP_AWB_IN_OUT_ATTR_S stInOrOut;
  CVI_BOOL bMultiLightSourceEn;
  ISP_AWB_MULTI_LS_TYPE_E enMultiLSType;
  CVI_U16 u16MultiLSScaler;
  CVI_U16 au16MultiCTBin[AWB_CT_BIN_NUM];
  CVI_U16 au16MultiCTWt[AWB_CT_BIN_NUM];
  CVI_BOOL bFineTunEn;
  CVI_U8 u8FineTunStrength;
  //AWB Algo 6
  struct ST_ISP_AWB_INTERFERNCE_S stInterfernce;
  struct ST_ISP_AWB_SKIN_S stSkin;
  struct ST_ISP_AWB_SKY_S stSky;
  struct ST_ISP_AWB_GRASS_S stGrass;
  struct ST_ISP_AWB_CT_WGT_S stCtLv;
  struct ST_ISP_AWB_SHIFT_LV_S stShiftLv;
  struct ST_ISP_AWB_REGION_S stRegion;
  CVI_U8 adjBgainMode;
  CVI_U8 reserve[239];
} ISP_AWB_ATTR_EX_S;

The macro is defined as follows:
#define AWB_CT_BIN_NUM    (8)
#define AWB_LS_NUM   (4)
```

【Member】

Member	Description
u8Tolerance	AWB adjust the deviation range. When the error is within this range, AWB will not adjust.
u8ZoneRadius	The size of AWB statistical partition. The smaller the value, the higher the accuracy of AWB, but it will reduce the stability of AWB algorithm
u16CurveLLimit	Left limit of AWB color temperature curve (R/G, B/G), value range: [0x0, 0x200]
u16CurveRLimit	Right limit of AWB color temperature curve (R/G, B/G), value range: [0x200, 0x3FF]
bExtraLightEn	Whether to turn on the independent light source
stLightInfo	Whether to consider the independent light source points outside the color temperature curve in AWB calculation, up to four independent points
stInOrOut	Parameters for AWB to judge indoor and outdoor scenes
bMultiLightSourceEn	AWB detects whether the current scene is a mixed light source to adjust the saturation or CCM
enMultiLSType	Adjust saturation or CCM
u16MultiLSScaler	Adjust the saturation or CCM intensity under mixed light source
au16MultiCTBin	The segmentation parameter of color temperature. Must be an increasing sequence
au16MultiCTWt	Color temperature segmentation weight, range: [0x0, 0x400]
bFineTunEn	AWB special color detection switch, such as skin color
u8FineTunStrength	The intensity of skin color, blue and other special color detection

【Note】

ISP_AWB_ATTR_EX_S extended attribute only works when pstWBAttr->enAlgType is AWB_ALG_ADVANCE

u8Tolerance is the sensitivity parameter of AWB.

When the value is 0, the AWB parameter will be updated imAWB/mediately.

When the value is relatively large, the AWB parameter will not be updated when the ambient color temperature slightly changes, and the screen will be slightly biased. It is recommended that the outdoor setting be 0 and the indoor setting be 2.

When u16CurveLLimit value < 512, it is mainly to exclude green objects, whereas when u16CurveRLimit value > 512, it is to exclude purplish areas

After bMultiLightSourceEn is turned on, AWB will judge whether the scene is a mixed light source. In such a scene, the saturation or CCM will be reduced to reduce the color deviation

In WDR mode, the mixed light detection function will be automatically turned off

【Related Data Type and Interface】

None

5.5.4.2 ISP_AWB_EXTRA_LIGHTSOURCE_INFO_S

【Description】

Define the parameters of the independent light point

【Syntax】

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

【Member】

Member	Description
u16WhiteRgain	R-channel gain of special light source
u16WhiteBgain	B-channel gain of special light source
u16ExpQuant	Judging by the external brightness. ExpQuant is the limit value of the brightness that is turned on. For example ExpQuant = 6, which means that the WB light source point is turned on below LV6 (the general night scene is below LV6) ExpQuant =106 means LV6 and above open ExpQuant =112 means it is enabled above LV12 (LV12 is generally outdoor)
u8LightStatus	Types of special light sources,0: no action 1: Add light point 2: Delete calculations near light points
u8Radius	Area size of special light source point, value range: [0x0, 0xFF]

【Note】

None

【Related Data Type and Interface】

None

5.5.4.3 ISP_AWB_IN_OUT_ATTR_S

【Description】

Define the parameters of AWB for outdoor and indoor judgment of the scene

【Syntax】

```
typedef struct _ISP_AWB_IN_OUT_ATTR_S {
  CVI_BOOL bEnable;
  ISP_OP_TYPE_E enOpType;
```

(continues on next page)

(continued from previous page)

```
  ISP_AWB_INDOOR_OUTDOOR_STATUS_E enOutdoorStatus;
  CVI_U32 u32OutThresh;
  CVI_U16 u16LowStart;
  CVI_U16 u16LowStop;
  CVI_U16 u16HighStart;
  CVI_U16 u16HighStop;
  CVI_BOOL bGreenEnhanceEn;
  CVI_U u8OutShiftLimit;
} ISP_AWB_IN_OUT_ATTR_S;
```

【Member】

Member	Description
bEnable	Judge whether the scene is indoor or outdoor switch
enOpType	Judge indoor and outdoor (automatic or manual)
enOutdoorStatus	Indoor or outdoor mode (in manual mode)
u32OutThresh	Threshold to determine indoor and outdoor. When the brightness is less than the threshold, it is judged that the LV of indoor and outdoor is more than 15.
u16LowStart	Lower the weight of low color temperature, and the starting point of low color temperature region is suggested to be 5000K.
u16LowStop	Lower the weight of low color temperature, and the end point of low color temperature area is recommended to be 4500k. Value range: (0,u16LowStart)
u16HighStart	Lower the weight of high color temperature, the starting point of high color temperature region is suggested to be 6500k. Value range: (u16LowStart, 0xFFFF]
u16HighStop	Lower the weight of high color temperature, and the end point of high color temperature area is recommended to be 8000K. Value range: (u16HighStart, 0xFFFF]
bGreenEnhanceEn	Switch for adding green channel in green plant scene
u8OutShiftLimit	When the scene is judged as outdoor scene, the AWB algorithm is limited by the range of white points

【Note】

Most outdoor brightness LV is greater than 15, so u32OutThresh is recommended to be greater than 15.

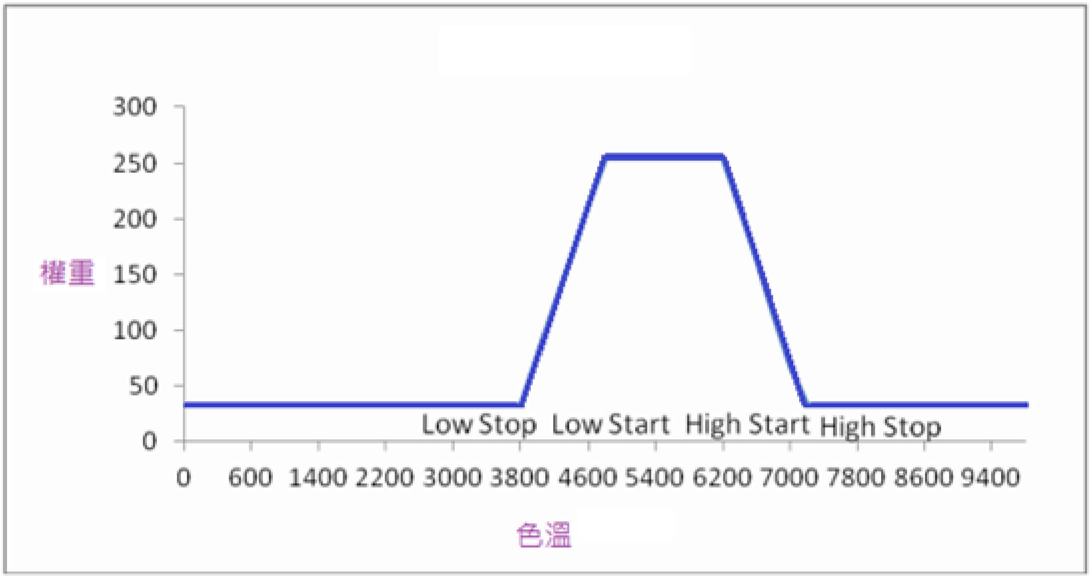
The range of the four parameters is as follows: u16LowStop<u16LowStart<u16HighStart<u16HighStop

【Related Data Type and Interface】

As shown in the figure below, u16LowStop is 3800K, and u16LowStart is 5000K

u16HighStart is 6200K, and u16HighStop is 7200K

The general weight is 32, and the maximum weight of outdoor color temperature is 256.



5.5.4.4 ISP_AWB_MULTI_LS_TYPE_E

【Description】

Define AWB strategy under mixed light source

【Syntax】

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

【Member】

Member	Description
AWB_MULTI_LS_SAT	Adjusting saturation under mixed light source
AWB_MULTI_LS_CCM	Adjusting CCM under mixed light source
AWB_MULTI_LS_BUTT	Invalid

【Note】

None

【Related Data Type and Interface】

None

5.5.4.5 ISP_AWB_INDOOR_OUTDOOR_STATUS_E

【Description】

Define AWB indoor and outdoor state

【Syntax】

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

【Member】

Member	Description
AWB_INDOOR_MODE	indoor mode
AWB_OUTDOOR_MODE	Outdoor mode
AWB_INDOOR_OUTDOOR_BUTT	invalid

【Note】

None

【Related Data Type and Interface】

None

5.5.5 ISP_WB_INFO_S

【Description】

Define white balance, saturation, color correction information

【Syntax】

```
typedef struct _ISP_WB_INFO_S {
  CVI_U16 u16Rgain;
  CVI_U16 u16Grgain;
  CVI_U16 u16Gbgain;
  CVI_U16 u16Bgain;
  CVI_U16 u16Saturation;
  CVI_U16 u16ColorTemp;
  CVI_U16 au16CCM[CCM_MATRIX_SIZE];
  CVI_U16 u16LS0CT;
  CVI_U16 u16LS1CT;
  CVI_U16 u16LS0Area;
  CVI_U16 u16LS1Area;
  CVI_U8  u8MultiDegree;
  CVI_U16 u16ActiveShift;
  CVI_U32 u32FirstStableTime;
```

(continues on next page)

(continued from previous page)

```
  ISP_AWB_INDOOR_OUTDOOR_STATUS_E enInOutStatus;
  CVI_S16 s16Bv;
} ISP_WB_INFO_S;
```

【Member】

Member	Description
u16Rgain	Current R-channel gain value
u16Grgain	Current Gr channel gain value
u16Gbgain	Current Gb channel gain value
u16Bgain	Current B channel gain value
u16Saturation	Current Saturation
u16ColorTemp	The current calculated color temperature value
au16CCM	The current color correction matrix, 10bit decimal precision. bit 15 is the sign bit, 0 means a positive number, 1 means a negative number, for example 0x8010 means -16
u16LS0CT	Color temperature of main light source in mixed light scene
u16LS1CT	Color temperature of secondary light source in mixed light scene
u16LS0Area	Main light area of mixed light scene
u16LS1Area	Secondary light area of mixed light scene
u8MultiDegree	The probability that the current scene is a mixed light source
u16ActiveShift	Limit value of current white balance range
u32FirstStableTime	The first AWB convergence time in frames
ISP_AWB_INDOOR_OUTDOOR_STATUS_E	Indoor and outdoor detection results
s16Bv	bv value of current environment

【Note】

None

【Related Data Type and Interface】

None

6 IMP

7 BlackLevel

7.1 Function Overview

Black level is the brightness value that the sensor still outputs when there is no light from outside. ISP needs to be deducted to make brightness and color normal.

7.2 API Reference

- *CVI_ISP_SetBlackLevelAttr*: Set the black level attribute parameters
- *CVI_ISP_GetBlackLevelAttr*: Get the black level attribute parameters

7.2.1 CVI_ISP_SetBlackLevelAttr

【Description】

Set the black level attribute parameters

【Syntax】

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

【Parameter】

Parameter	Description	Input/ Output
ViPipe	VI_PIPE number	Input
pstBlackLevelAttr	Black level attribute parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_GetBlackLevelAttr*

7.2.2 CVI_ISP_GetBlackLevelAttr

【Description】

Get the black level attribute parameters

【Syntax】

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

【Parameter】

Parameter	Description	Input/ Output
ViPipe	VI_PIPE number	Input
pstBlackLevelAttr	Black level attribute parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_SetBlackLevelAttr*

7.3 Data Types

- *ISP_BLACK_LEVEL_MANUAL_ATTR_S*: Define black level manual attributes
- *ISP_BLACK_LEVEL_AUTO_ATTR_S*: Define black level auto attributes
- *ISP_BLACK_LEVEL_ATTR_S*: Define black level attributes

7.3.1 ISP_BlACK_LEVEL_MANUAL_ATTR_S

【Description】

Define black level manual attributes

【定义】

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

【成员】

Member	Description
OffsetR	BLC R pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGr	BLC GR pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGb	BLC GB pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetB	BLC B pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetR2	BLC second stage R pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGr2	BLC second level GR pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGb2	BLC second level GB pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetB2	BLC second level B pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetBlackLevelAttr*
- *CVI_ISP_GetBlackLevelAttr*

7.3.2 ISP_BLACK_LEVEL_AUTO_ATTR_S

【Description】

Define black level auto attributes

【定义】

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

(continues on next page)

(continued from previous page)

```
  CVI_U16 OffsetGr2[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 OffsetGb2[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 OffsetB2[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_BLACK_LEVEL_AUTO_ATTR_S;
```

【成员】

Member	Description
OffsetR	BLC R pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGr	BLC GR pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGb	BLC GB pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetB	BLC B pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetR2	BLC second stage R pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGr2	BLC second level GR pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetGb2	BLC second level GB pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16
OffsetB2	BLC second level B pixel dark current value Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetBlackLevelAttr*
- *CVI_ISP_GetBlackLevelAttr*

7.3.3 ISP_BLACK_LEVEL_ATTR_S

【Description】

Define black level attributes

【定义】

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

【成员】

Member	Description
Enable	The BLC module enable Value range: [0x0, 0x1] Data type: CVI_U8
enOpType	Choose manual or automatic mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
stManual	manual parameters
stAuto	automatic parameters

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetBlackLevelAttr*
- *CVI_ISP_GetBlackLevelAttr*

8 DPC

8.1 Function Overview

DPC, the abbreviation for Defect Pixel Correction, can fix the bad points of problems on Sensor.

There are two methods of correction, static correction and dynamic correction.

- Static correction:The correction can fall into two categories, dark spot correction and bright spot correction. In the bright spot correction, the lens is blacked out and the bad point calibration procedure is started. In the dark spot correction, when the background is flat, such as before the gray box, adjust the exposure to make the overall brightness of the image about 50%. The static bad points allow no more than 4095 bad points.
- Dynamic correction: Under this method, the bad points are judged dynamically and corrected rather than using the correction data. In low noise situation, it is helpful for color deviation, but if the intensity is too strong, the details of the picture may be reduced.

8.2 API Reference

- *CVI_ISP_SetDPDynamicAttr*: Set dynamic bad point correction attributes
- *CVI_ISP_GetDPDynamicAttr*: Get dynamic bad point correction attributes
- *CVI_ISP_SetDPCalibrate*: Set static bad point calibration parameters
- *CVI_ISP_GetDPCalibrate*: Get static bad point calibration parameters
- *CVI_ISP_SetDPStaticAttr*: Set static bad point correction attributes
- *CVI_ISP_GetDPStaticAttr*: Get static bad point correction attributes

8.2.1 CVI_ISP_SetDPDynamicAttr

【Description】

Set dynamic bad point correction attributes.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPCDynamicAttr	Dynamic bad point correction attribute	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_GetDPDynamicAttr*

8.2.2 CVI_ISP_GetDPDynamicAttr

【Description】

Get dynamic bad point correction attributes

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPDynamicAttr	Static bad point calibration parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_SetDPDynamicAttr*

8.2.3 CVI_ISP_SetDPCalibrate

【Description】

Set static bad point calibration parameters

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPCalibrateAttr	Static bad point calibration parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_GetDPCalibrate*

8.2.4 CVI_ISP_GetDPCalibrate

【Description】

Get calibration parameters of static bad points

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPCalibrateAttr	Static bad point calibration parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related Topic】

- *CVI_ISP_SetDPCalibrate*

8.2.5 CVI_ISP_SetDPStaticAttr

【Description】

Set static bad point correction attribute.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPStaticAttr	Static bad point calibration parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDPStaticAttr*

8.2.6 CVI_ISP_GetDPStaticAttr

【Description】

Get static bad point correction attribute.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstDPStaticAttr	Static bad point calibration parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDPStaticAttr*

8.3 Data Types

- *ISP_DP_DYNAMIC_MANUAL_ATTR_S*: Manual attribute of dynamic bad point correction
- *ISP_DP_DYNAMIC_AUTO_ATTR_S*: Automatic attributes of dynamic bad point correction
- *ISP_DP_DYNAMIC_ATTR_S*: Dynamic bad point correction attribute
- *ISP_DP_CALIB_ATTR_S*: Static bad point calibration parameters

- *ISP_DP_STATIC_ATTR_S*: Static bad point correction attribute

8.3.1 ISP_DP_DYNAMIC_MANUAL_ATTR_S

【Description】

Manual attribute of dynamic bad point correction.

【Syntax】

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

【Member】

Member	Description
ClusterSize	the upper limit of the area of clustering bad points; the higher the value is, the better the correction effect is, but it may cause the attenuation of resolution in high frequency region Value range: [0x0, 0x3] Data type: CVI_U8
BrightDefectToNormalPixRatio	The ratio between the value of visible bright and bad points and the surrounding pixels (Q4.4) Value range: [0x0, 0xff] Data type: CVI_U8
DarkDefectToNormalPixRatio	The ratio between the value of visible dark and bad points and the surrounding pixels (Q4.4) Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreR	critical value of R-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreG	critical value of G-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreB	critical value of B-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreMinG	the minimum critical value of g-channel judging flat area Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreMinRB	the minimum critical value of RB-channel judging flat area Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDPDynamicAttr*
- *CVI_ISP_GetDPDynamicAttr*

8.3.2 ISP_DP_DYNAMIC_AUTO_ATTR_S

【Description】

Automatic attributes of dynamic bad point correction.

【Syntax】

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

【Member】

Member	Description
ClusterSize	the upper limit of the area of clustering bad points; the higher the value is, the better the correction effect is, but it may cause the attenuation of resolution in high frequency region Value range: [0x0, 0x3] Data type: CVI_U8
BrightDefectToNormalPixRatio	The ratio between the value of visible bright and bad points and the surrounding pixels (Q4.4) Value range: [0x0, 0xff] Data type: CVI_U8
DarkDefectToNormalPixRatio	The ratio between the value of visible dark and bad points and the surrounding pixels (Q4.4) Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreR	critical value of R-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreG	critical value of G-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreB	critical value of B-channel in flat area;the smaller the critical value, the more edge information can be preserved Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreMinG	the minimum critical value of g-channel judging flat area Value range: [0x0, 0xff] Data type: CVI_U8
FlatThreMinRB	the minimum critical value of RB-channel judging flat area Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDPDynamicAttr*
- *CVI_ISP_GetDPDynamicAttr*

8.3.3 ISP_DP_DYNAMIC_ATTR_S

【Description】

Dynamic bad point correction attribute.

【Syntax】

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

【Member】

Member	Description
Enable	The DPC module enable 0: off 1: enable Value range: [0, 1] Data type: CVI_BOOL
DynamicDPCEnable	Enable dynamic dead point correction function Data type: CVI_U32
enOpType	Operation type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Parameters in manual mode
stAuto	Parameters in automatic mode

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDPDynamicAttr*
- *CVI_ISP_GetDPDynamicAttr*

8.3.4 ISP_DP_CALIB_ATTR_S

【Description】

Static bad point calibration parameters.

【Syntax】

```
typedef struct _ISP_DP_CALIB_ATTR_S {
  CVI_BOOL EnableDetect;
  CVI_STATIC_DP_TYPE_E StaticDPType;
  CVI_U8 StartThresh;
  CVI_U16 CountMax;
  CVI_U16 CountMin;
  CVI_U16 TimeLimit;
  CVI_BOOL saveFileEn;

  // read only
  CVI_U32 Table[STATIC_DP_COUNT_MAX];
  CVI_U8 FinishThresh;
  CVI_U16 Count;
  ISP_STATUS_E Status;
} ISP_DP_CALIB_ATTR_S;
```

【Member】

Member	Description
EnableDetect	Static bad point calibration enablement; Value range: [0, 1] Data type: CVI_BOOL
StaticDPType	Static bad point calibration type; Value range: [0x0, 0x1] Data type: CVI_STATIC_DP_TYPE_E
StartThresh	Detection threshold at the beginning of static bad point calibration; Value range: [0x0, 0xff] Data type: CVI_U8
CountMax	Maximum number of allowed static bad points; Value range: [0x0, 0xfff] Data type: CVI_U16
CountMin	Minimum number of allowed static bad points; Value range: [0x0, 0xfff] Data type: CVI_U16
TimeLimit	Allowable calibration timeout threshold; Value range: [0x0, 0x640] Data type: CVI_U16
saveFileEn	Whether to save the raw image Value range: [0, 1] Data type: CVI_BOOL
Table[4096]	Read only, look-up table of bright and dark bad point coordinate values; low 29bit is valid, [12:0] bit is the horizontal coordinate of the bad point, and [28:16] bit is the vertical coordinate of the bad point; Value range: [0x0, 0x1ff1fff] Data type: CVI_U32
FinishThresh	Read only, the detection threshold at the end of static bad point calibration; Value range: [0x0, 0xff] Data type: CVI_U8
Count	Read only, the number of calibrated static bad points; Value range: [0x0, 0xfff] Data type: CVI_U16
Status	Read only, static bad point calibration result status information; Value range: [0x0, 0x2] Data type: ISP_STATUS_E

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDPCalibrate*
- *CVI_ISP_GetDPCalibrate*

8.3.5 ISP_DP_STATIC_ATTR_S

【Description】

Static bad point calibration attributes.

【Syntax】

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

【Member】

Member	Description
Enable	DPC module enablement; 0: close 1: enable Value range: [0, 1] Data type: CVI_BOOL
BrightCount	Number of bright and bad points; Value range: [0x0, 0xfff] Data type: CVI_U16
DarkCount	Number of dark and bad point; Value range: [0x0, 0xfff] Data type: CVI_U16
BrightTable[4095]	Coordinate information of bright and bad points; the low 29bit is valid, the [12:0] bit is the horizontal coordinate of the bad point, and the [28:16] bit is the vertical coordinate of the bad point. Value range: [0x0, 0x1ff1fff] Data type: CVI_U32
DarkTable[4095]	the coordinate value of dark point; the low 29bit is effective, [12:0] bit is the horizontal coordinates of the bad point, and [28:16] bit is the vertical coordinates of the bad point. Value range: [0x0, 0x1ff1fff] Data type: CVI_U32
Show	Static bad point display enablement is not supported yet. Value range: [0, 1] Data type: CVI_BOOL

【Note】

None.

- *CVI_ISP_SetDPStaticAttr*
- *CVI_ISP_GetDPStaticAttr*

9 MeshShading

9.1 Function Description

- The dark corners around the lens are corrected in the form of grid

9.2 API Reference

- *CVI_ISP_SetMeshShadingAttr*: Set Mesh Shading algorithm parameters.
- *CVI_ISP_GetMeshShadingAttr*: Get Mesh Shading algorithm parameters.
- *CVI_ISP_SetMeshShadingGainLutAttr*: Set LSC grid form compensation gain table.
- *CVI_ISP_GetMeshShadingGainLutAttr*: Get LSC grid form compensation gain table.

9.2.1 CVI_ISP_SetMeshShadingAttr

【Description】

Set Mesh Shading algorithm parameters.

【Syntax】

```
CVI_S32 CVI_ISP_SetMeshShadingAttr(VI_PIPE ViPipe, const ISP_MESH_SHADING_ATTR_
↪S *pstMeshShadingAttr);
```

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstMeshShadingAttr	Mesh Shading algorithm parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetMeshShadingAttr*

9.2.2 CVI_ISP_GetMeshShadingAttr

【Description】

Get Mesh Shading algorithm parameters.

【Syntax】

```
CVI_S32 CVI_ISP_GetMeshShadingAttr(VI_PIPE ViPipe, ISP_MESH_SHADING_ATTR_S
*pstMeshShadingAttr);
```

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstMeshShadingAttr	Mesh Shading algorithm parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetMeshShadingAttr*

9.2.3 CVI_ISP_SetMeshShadingGainLutAttr

【Description】

Set LSC grid form compensation gain table.

【Syntax】

```
CVI_S32 CVI_ISP_SetMeshShadingGainLutAttr(VI_PIPE ViPipe, const ISP_MESH_
↪SHADING_GAIN_LUT_ATTR_S *pstMeshShadingGainLutAttr);
```

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstMeshShading-GainLutAttr	LSC grid form compensation gain table	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetMeshShadingGainLutAttr*

9.2.4 CVI_ISP_GetMeshShadingGainLutAttr

【Description】

Get LSC grid form compensation gain table.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstMeshShading-GainLutAttr	LSC grid form compensation gain table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetMeshShadingGainLutAttr*

9.3 Data Types

- *ISP_MESH_SHADING_MANUAL_ATTR_S*: Manual parameters of Mesh Shading algorithm.
- *ISP_MESH_SHADING_AUTO_ATTR_S*: Automated parameters of Mesh Shading algorithm.
- *ISP_MESH_SHADING_ATTR_S*: Mesh Shading algorithm parameters.
- *ISP_MESH_SHADING_GAIN_LUT_S*: LSC grid form compensation gain table details.

- *ISP_MESH_SHADING_GAIN_LUT_ATTR_S*: LSC grid form compensation gain table.

9.3.1 ISP_MESH_SHADING_MANUAL_ATTR_S

【Description】

Manual parameters of Mesh Shading algorithm.

【Syntax】

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

【Member】

Member	Description
MeshStr	LSC compensation strength; Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMeshShadingAttr*
- *CVI_ISP_GetMeshShadingAttr*

9.3.2 ISP_MESH_SHADING_AUTO_ATTR_S

【Description】

Automated parameters of Mesh Shading algorithm.

【Syntax】

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

【Member】

Member	Description
MeshStr	LSC compensation strength; Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMeshShadingAttr*
- *CVI_ISP_GetMeshShadingAttr*

9.3.3 ISP_MESH_SHADING_ATTR_S

【Description】

Mesh Shading algorithm parameters.

【Syntax】

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

【Member】

Member	Description
Enable	Enable LSC function. 0: close. 1: enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	Select manual or automatic mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
OverflowProtection	Avoid color cast in overexposed areas Value range: [0, 1] Data type: CVI_BOOL
stManual	Manual parameters
stAuto	Automatic parameters

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMeshShadingAttr*
- *CVI_ISP_GetMeshShadingAttr*

9.3.4 ISP_MESH_SHADING_GAIN_LUT_S

【Description】

LSC grid form compensation gain table details.

【Syntax】

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

【Member】

Member	Description
ColorTemperature	the color temperature corresponding to the color temperature adaptive LSC compensation gain table, in K. Value range: [0x0, 0x7530] Data type: CVI_U16
RGain	Red channel gain; Value range: [0x0, 0xfff] Data type: CVI_U16
GGain	Green channel gain; Value range: [0x0, 0xfff] Data type: CVI_U16
BGain	Blue channel gain; Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMeshShadingGainLutAttr*
- *CVI_ISP_GetMeshShadingGainLutAttr*

9.3.5 ISP_MESH_SHADING_GAIN_LUT_ATTR_S

【Description】

LSC grid form compensation gain table.

【Syntax】

```
typedef struct _ISP_MESH_SHADING_GAIN_LUT_ATTR_S {
   CVI_U8 Size;
   ISP_MESH_SHADING_GAIN_LUT_S LscGainLut[ISP_MLSC_COLOR_TEMPERATURE_SIZE];
} ISP_MESH_SHADING_GAIN_LUT_ATTR_S;
```

【Member】

Member	Description
Size	Number of color temperature adaptive LSC compensation gain meters; Value range: [0x1, 0x7] Data type: CVI_U8
LscGainLut	LSC grid form compensation gain table

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMeshShadingGainLutAttr*
- *CVI_ISP_GetMeshShadingGainLutAttr*

10 RadialShading

10.1 Function Overview

Correct the vignetting around the lens in the form of concentric circles

10.2 API Reference

- *CVI_ISP_SetRadialShadingAttr* : Set Radial LSC parameters
- *CVI_ISP_GetRadialShadingAttr* : Get Radial LSC parameters
- *CVI_ISP_SetRadialShadingGainLutAttr* : Set the LSC Radius form compensation gain table
- *CVI_ISP_GetRadialShadingGainLutAttr* : Get the LSC Radius form compensation gain table

10.2.1 CVI_ISP_SetRadialShadingAttr

【Description】

Set Radial LSC parameters

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstRadialShadingAttr	Mesh Shading algorithm parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

This function of cv180x & cv181x is not supported.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetRadialShadingAttr*

10.2.2 CVI_ISP_GetRadialShadingAttr

【Description】

- Get Radial LSC parameters

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstRadialShadingAttr	Radial LSC parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

This function of cv180x & cv181x is not supported.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetRadialShadingAttr*

10.2.3 CVI_ISP_SetRadialShadingGainLutAttr

【Description】

- Set the LSC Radius form compensation gain table

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstRadialShading-GainLutAttr	LSC Radius Form Compensation Gain Table	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

This function of cv180x & cv181x is not supported.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetRadialShadingGainLutAttr*

10.2.4 CVI_ISP_GetRadialShadingGainLutAttr

【Description】

Get the LSC Radius form compensation gain table

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstRadialShading-GainLutAttr	LSC Radius Form Compensation Gain Table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

This function of cv180x & cv181x is not supported.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetRadialShadingGainLutAttr*

10.3 Data Types

- *ISP_RADIAL_SHADING_MANUAL_ATTR_S* : Radial LSC manual parameter
- *ISP_RADIAL_SHADING_AUTO_ATTR_S* : Radial LSC automatic parameter
- *ISP_RADIAL_SHADING_ATTR_S* : Radial LSC parameter
- *ISP_RADIAL_SHADING_GAIN_LUT_S* : Radial LSC Radius form compensation gain table item

- *ISP_RADIAL_SHADING_GAIN_LUT_ATTR_S* : LSC Radius form compensation gain table

10.3.1 ISP_RADIAL_SHADING_MANUAL_ATTR_S

【Description】

- Radial LSC manual parameter

【Syntax】

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

【Member】

Member	Description
RadiusStr	LSC compensation strength Value range: [0x0, 0xfff] Data Type: CVI_U16
RadiusIRStr	LSC IR Compensation Strength Value range: [0x0, 0xfff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRadialShadingAttr*
- *CVI_ISP_GetRadialShadingAttr*

10.3.2 ISP_RADIAL_SHADING_AUTO_ATTR_S

【Description】

Radial LSC automatic parameter

【Syntax】

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

【Member】

Member	Description
RadiusStr	LSC compensation strength Value range: [0x0, 0xfff] Data Type: CVI_U16
RadiusIRStr	LSC IR Compensation Strength Value range: [0x0, 0xfff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRadialShadingAttr*
- *CVI_ISP_GetRadialShadingAttr*

10.3.3 ISP_RADIAL_SHADING_ATTR_S

【Description】

- Radial LSC parameter

【Syntax】

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

【Member】

Member	Description
Enable	The LSC function is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
enOpType	Choose manual or automatic mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data Type: CVI_U8
CenterX	X-direction coordinates of image sensor mirror center Value range: [0x0, 0x1fff] Data Type: CVI_U16
CenterY	Y-direction coordinates of image sensor mirror center Value range: [0x0, 0x1fff] Data Type: CVI_U16
RadiusScaleRGB	RGB Radius and compensation gain table normalization coefficient Value range: [0x0, 0x7fff] Data Type: CVI_U16
RadiusScaleIR	IR Radius and Compensation Gain Table Normalization Coefficients Value range: [0x0, 0x7fff] Data Type: CVI_U16
stManual	manual parameters
stAuto	automatic parameters

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRadialShadingAttr*
- *CVI_ISP_GetRadialShadingAttr*

10.3.4 ISP_RADIAL_SHADING_GAIN_LUT_S

【Description】

- Radial LSC Radius form compensation gain table item

【Syntax】

```
typedef struct _ISP_RADIAL_SHADING_GAIN_LUT_S {
    CVI_U16 ColorTemperature;
    CVI_U16 RGain[ISP_RLSC_WINDOW_SIZE];
    CVI_U16 GGain[ISP_RLSC_WINDOW_SIZE];
```

(continues on next page)

(continued from previous page)

```
    CVI_U16 BGain[ISP_RLSC_WINDOW_SIZE];
    CVI_U16 IrGain[ISP_RLSC_WINDOW_SIZE];
} ISP_RADIAL_SHADING_GAIN_LUT_S;
```

【Member】

Member	Description
ColorTemperature	The color temperature corresponding to the color temperature adaptive LSC compensation gain table, the unit is K Value range: [0x0, 0xffff] Data Type: CVI_U16
RGain	LSC Radius form red channel gain Value range: [0x0, 0xfff] Data Type: CVI_U16
GGain	LSC Radius form green channel gain Value range: [0x0, 0xfff] Data Type: CVI_U16
BGain	LSC Radius form blue channel gain Value range: [0x0, 0xfff] Data Type: CVI_U16
IrGain	LSC Radius form IR color channel gain Value range: [0x0, 0xfff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRadialShadingGainLutAttr*
- *CVI_ISP_GetRadialShadingGainLutAttr*

10.3.5 ISP_RADIAL_SHADING_GAIN_LUT_ATTR_S

【Description】

- LSC Radius form compensation gain table

【Syntax】

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

【Member】

Member	Description
Size	Color temperature adaptive LSC compensation gain table quantity Value range: [0x1, 0x7] Data Type: CVI_U8
ColorTemperature	The color temperature corresponding to the color temperature adaptive LSC compensation gain table, the unit is K Value range: [0x0, 0xffff] Data Type: CVI_U16
RLscGainLut	LSC Radius Form Compensation Gain Table Details Value range: [0x0, 0xfff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRadialShadingGainLutAttr*
- *CVI_ISP_GetRadialShadingGainLutAttr*

11 CCM

11.1 Function Overview

CCM is the acronym for color correction matrix.

Because the sensor’s response in RGB is not the same as that of human eye,

this matrix needs to be used for conversion,

so that the brightness of the image captured by the front end is consistent with that of human eye.

11.2 API Reference

- *CVI_ISP_SetCCMAttr*: Set attribute parameters of color correction matrix.
- *CVI_ISP_GetCCMAttr*: Get attribute parameters of color correction matrix.
- *CVI_ISP_SetCCMSaturationAttr*: Set attribute parameters of color Saturation correction matrix.
- *CVI_ISP_GetCCMSaturationAttr*: Get attribute parameters of color Saturation correction matrix.

11.2.1 CVI_ISP_SetCCMAttr

【Description】

Set attribute parameters of color correction matrix.

【Syntax】

```
CVI_S32 CVI_ISP_SetCCMAttr(VI_PIPE ViPipe, const ISP_CCM_ATTR_S *pstCCMAttr);
```

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstCCMAttr	attribute parameters of color correction matrix	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCCMAttr*

11.2.2 CVI_ISP_GetCCMAttr

【Description】

Get attribute parameters of color correction matrix.

【Syntax】

```
CVI_S32 CVI_ISP_GetCCMAttr(VI_PIPE ViPipe, ISP_CCM_ATTR_S *pstCCMAttr);
```

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstCCMAttr	attribute parameters of color correction matrix	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCCMAttr*

11.2.3 CVI_ISP_SetCCMSaturationAttr

【Description】

Set color saturation property parameters

【Syntax】

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

【Parameter】

参数名称	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstCCMSaturationAttr	Color saturation attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCCMSaturationAttr*

11.2.4 CVI_ISP_GetCCMSaturationAttr

【Description】

Get color saturation property parameters

【Syntax】

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

【Parameter】

参数名称	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstCCMSaturationAttr	Color saturation attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCCMSaturationAttr*

11.3 Data Types

- *ISP_COLORMATRIX_ATTR_S* : Color correction matrix
- *ISP_CCM_MANUAL_ATTR_S* : Color correction matrix attribute manual parameters
- *ISP_CCM_AUTO_ATTR_S* : Color correction matrix attribute automatic parameter
- *ISP_CCM_ATTR_S* : Color correction matrix attribute parameters
- *ISP_CCM_SATURATION_MANUAL_ATTR_S* : Color saturation attribute manual parameter
- *ISP_CCM_SATURATION_AUTO_ATTR_S* : Color saturation attribute automatic parameter
- *ISP_CCM_SATURATION_ATTR_S* : Color saturation attribute parameter

11.3.1 ISP_COLORMATRIX_ATTR_S

【Description】

- Color correction matrix

【Syntax】

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

【Member】

Member	Description
ColorTemp	the color temperature of the color correction matrix; Value range: [0x1f4, 0x7530] Data type: CVI_U16
CCM[9]	Actual content of 3x3 color correction matrix; Value range: [0x-2000, 0x1fff] Data type: CVI_S16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMAttr*
- *CVI_ISP_GetCCMAttr*

11.3.2 ISP_CCM_MANUAL_ATTR_S

【Description】

Color correction matrix attribute manual parameters.

【Syntax】

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

【Member】

Member	Description
SatEnable	In manual mode, whether saturation is enabled. Value range: [0, 1] Data type: CVI_BOOL
CCM[9]	Value range: [0x-2000, 0x1fff] Data type: CVI_S16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMAttr*
- *CVI_ISP_GetCCMAttr*

11.3.3 ISP_CCM_AUTO_ATTR_S

【Description】

- Color correction matrix attribute automatic parameter

【Syntax】

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

【Member】

Member	Description
ISOActEnable	CCM Bypass function is enabled under low illumination. 0: close. 1: Enable. Value range: [0, 1] Data type: CVI_BOOL
TempActEnable	CCM bypass function is enabled at high and low color temperatures. 0: close. 1: Enable. Value range: [0, 1] Data type: CVI_BOOL
CCMTabNum	the number of CCM matrices currently configured. Value range: [0x3, 0x7] Data type: CVI_U8
CCMTab	Color correction matrix under different color temperature

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMAttr*
- *CVI_ISP_GetCCMAttr*

11.3.4 ISP_CCM_ATTR_S

【Description】

Color correction matrix attribute parameters

【Syntax】

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

【Member】

Member	Description
Enable	CCM module is enabled. 0: close. 1: Enable. Value range: [0, 1] Data type: CVI_BOOL
enOpType	Select manual or automatic mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
stAuto	Automatic parameters
stManual	Manual parameters

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMAttr*
- *CVI_ISP_GetCCMAttr*

11.3.5 ISP_CCM_SATURATION_MANUAL_ATTR_S

【Description】

Color saturation attribute manual parameter

【Syntax】

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

【Member】

Member	Description
SaturationLE	long exposure saturation Value range: [0, 0xff] Data type: CVI_U8
SaturationSE	short exposure saturation Value range: [0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMSaturationAttr*
- *CVI_ISP_GetCCMSaturationAttr*

11.3.6 ISP_CCM_SATURATION_AUTO_ATTR_S

【Description】

Color saturation attribute automatic parameter

【Syntax】

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

【Member】

Member	Description
SaturationLE	long exposure saturation Value range: [0, 0xff] Data type: CVI_U8
SaturationSE	short exposure saturation Value range: [0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMSaturationAttr*
- *CVI_ISP_GetCCMSaturationAttr*

11.3.7 ISP_CCM_SATURATION_ATTR_S

【Description】

Color saturation attribute parameter

【Syntax】

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

【Member】

Member	Description
stManual	manual parameters
stAuto	automatic parameters

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCCMSaturationAttr*
- *CVI_ISP_GetCCMSaturationAttr*

s.. vim: syntax=rst

12 Noise Profile

12.1 Function Overview

Record the noise of sensor gain. Generally, it is obtained by the calibration program and does not need special adjustment.

12.2 API Reference

- *CVI_ISP_SetNoiseProfileAttr*: Set Noise Profile attribute parameter.
- *CVI_ISP_GetNoiseProfileAttr*: Get Noise Profile attribute parameter.

12.2.1 CVI_ISP_SetNoiseProfileAttr

【Description】

Set Noise Profile attribute parameter.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstNoiseProfileAttr	Noise Profile attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetNoiseProfileAttr*

12.2.2 CVI_ISP_GetNoiseProfileAttr

【Description】

Get the Noise Profile property parameter.

【Syntax】

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

【Parameter】

Parameter	Description	Input / Output
ViPipe	VI_PIPE number	Input
pstNoiseProfileAttr	Noise Profile attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetNoiseProfileAttr*

12.3 Data Types

- *ISP_CMOS_NOISE_CALIBRATION_S*: Noise Profile attribute parameter.

12.3.1 ISP_CMOS_NOISE_CALIBRATION_S

【Description】

Noise Profile property parameter

【Syntax】

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

【Member】

Member	Description
CalibrationCoef [16][4][2]	Raw data noise model, divided into R/Gr/Gb/B channels, floating point data formats Value range： [-2147483648,2147483647] Data type： CVI_FLOAT

【Note】

none

【Related Data Type and Interface】

- *CVI_ISP_SetNoiseProfileAttr*
- *CVI_ISP_GetNoiseProfileAttr*

13 BNR

13.1 Function Description

Spatial denoising algorithm in Bayer Domain.

13.2 API Reference

- *CVI_ISP_SetNRAttr* : Set Bayer noise reduction parameter attributes
- *CVI_ISP_GetNRAttr* : Get Bayer noise reduction parameter attributes
- *CVI_ISP_SetNRFilterAttr* : Set the Bayer noise reduction filter attribute
- *CVI_ISP_GetNRFilterAttr* : Get Bayer noise reduction filter attributes

13.2.1 CVI_ISP_SetNRAttr

【Description】

Set Bayer noise reduction parameter attributes

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstNRAttr	Bayer denoising parameter properties	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetNRAttr*

13.2.2 CVI_ISP_GetNRAttr

【Description】

Get Bayer noise reduction parameter attributes

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstNRAttr	Bayer denoising parameter properties	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetNRAttr*

13.2.3 CVI_ISP_SetNRFilterAttr

【Description】

Set the Bayer noise reduction filter attribute

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstNRFilterAttr	Bayer denoising filter properties	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetNRFilterAttr*

13.2.4 CVI_ISP_GetNRFilterAttr

【Description】

Get Bayer noise reduction filter attributes

【Syntax】

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

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstNRFilterAttr	Bayer denoising filter properties	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetNRFilterAttr*

13.3 Data Types

- *ISP_NR_MANUAL_ATTR_S* : Bayer noise reduction parameter manual attribute
- *ISP_NR_AUTO_ATTR_S* : Bayer noise reduction parameter automatic attribute
- *ISP_NR_ATTR_S* : Bayer noise reduction parameter attribute
- *ISP_NR_FILTER_MANUAL_ATTR_S* : Bayer noise reduction filter manual attribute
- *ISP_NR_FILTER_AUTO_ATTR_S* : Bayer noise reduction filter automatic attribute
- *ISP_NR_FILTER_ATTR_S* : Bayer noise reduction filter attribute
- *ISP_RLSC_MANUAL_ATTR_S* : Bayer noise reduction RLSC manual attribute
- *ISP_RLSC_AUTO_ATTR_S* : Bayer noise reduction RLSC automatic attribute
- *ISP_RLSC_ATTR_S* : Bayer noise reduction RLSC attribute

13.3.1 ISP_NR_MANUAL_ATTR_S

【Description】

Bayer Noise Reduction Parameter Manual Properties

【Syntax】

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

【Member】

Member	Description
WindowType	De-noise filter localization.The smaller the value, the more local the effect. Value range: [0x0, 0xb] Data type: CVI_U8
DetailSmoothMode	Noise removal detail smoothing feature enabled. 0: Off 1: Enable Value range: [0x0, 0x1] Data type: CVI_U8
NoiseSuppressStr	Noise suppression intensity.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
FilterType	Denoising filter strength.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressStrMode	Intensity of bright noise and noise removal.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRAttr*
- *CVI_ISP_GetNRAttr*

13.3.2 ISP_NR_AUTO_ATTR_S

【Description】

Bayer Noise Reduction Parameter Auto Properties

【Syntax】

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

【Member】

Member	Description
WindowType	De-noise filter localization.The smaller the value, the more local the effect. Value range: [0x0, 0xb] Data type: CVI_U8
DetailSmoothMode	Noise removal detail smoothing feature enabled. 0: Off 1: Enable Value range: [0x0, 0x1] Data type: CVI_U8
NoiseSuppressStr	Noise suppression intensity.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
FilterType	Denoising filter strength.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressStrMode	Intensity of bright noise and noise removal.The greater the value, the greater the intensity of noise removal. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRAttr*

- *CVI_ISP_GetNRAttr*

13.3.3 ISP_NR_ATTR_S

【Description】

Bayer noise reduction parameter attribute

【Syntax】

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

【Member】

Member	Description
Enable	The BNR module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
enOpType	Operating mode 0: automatic 1: Manual
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
CoringParamEnable	Coring parameter enable Value range: [0, 1] Data Type: CVI_BOOL
stManual	Bayer denoising parameters manual properties
stAuto	Bayer denoising parameters automatic properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRAttr*
- *CVI_ISP_GetNRAttr*

13.3.4 ISP_NR_FILTER_MANUAL_ATTR_S

【Description】

- Bayer noise reduction filter manual attribute

【Syntax】

```
typedef struct _ISP_NR_FILTER_MANUAL_ATTR_S {
  CVI_U8 LumaStr[8];
  CVI_U8 VarThr;
  CVI_U16 CoringWgtLF;
  CVI_U16 CoringWgtHF;
  CVI_U8 NonDirFiltStr;
  CVI_U8 VhDirFiltStr;
  CVI_U8 AaDirFiltStr;
  CVI_U16 NpSlopeR; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpSlopeGr; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpSlopeGb; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpSlopeB; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLumaThrR; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLumaThrGr; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLumaThrGb; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLumaThrB; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLowOffsetR; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLowOffsetGr; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLowOffsetGb; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpLowOffsetB; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpHighOffsetR; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpHighOffsetGr; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpHighOffsetGb; /*RW; Range:[0x0, 0x3ff]*/
  CVI_U16 NpHighOffsetB; /*RW; Range:[0x0, 0x3ff]*/
} ISP_NR_FILTER_MANUAL_ATTR_S;
```

【Member】

Member	Description
LumaStr[8]	The reference image brightness adjusts the denoising strength respectively from dark to bright. The larger the value, the stronger the bright noise removal. Value range: [0x0, 0x1f] Data type: CVI_U8
VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged. Value range: [0x0, 0xff] Data type: CVI_U8
CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region. Value range: [0x0, 0x100] Data type: CVI_U16
CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region. Value range: [0x0, 0x100] Data type: CVI_U16
NonDirFiltStr	Adjusts the strength of noise reduction in the low frequency region. The larger the value, the more noise is removed in the low frequency area. Value range: [0x0, 0x1f] Data type: CVI_U8
VhDirFiltStr	Adjusts the strength of denoising in the horizontal and vertical areas. Larger values remove more noise on horizontal and vertical edges. Value range: [0x0, 0x1f] Data type: CVI_U8
AaDirFiltStr	Adjusts the strength of denoising on diagonal edges. The larger the value, the more noise is removed on the diagonal edges. Value range: [0x0, 0x1f] Data type: CVI_U8
NpSlopeR	The slope of the Noise profile in the R channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpSlopeGr	The slope of the Noise profile in the Gr channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpSlopeGb	Noise profile slope in Gb channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpSlopeB	The slope of the Noise profile on the B channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpLumaThrR	Luminance threshold of the Noise profile in the R channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpLumaThrGr	Luminance threshold of the Noise profile in the Gr channel Value range: [0x0, 0x3ff] Data type: CVI_U8
NpLumaThrGb	Luminance threshold of Noise profile in Gb channel

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRFilterAttr*
- *CVI_ISP_GetNRFilterAttr*

13.3.5 ISP_NR_FILTER_AUTO_ATTR_S

【Description】

Bayer noise reduction filter automatic attribute

【Syntax】

```
typedef struct _ISP_NR_FILTER_AUTO_ATTR_S {
  CVI_U8 LumaStr[8][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 VarThr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 CoringWgtLF[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 CoringWgtHF[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NonDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 VhDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 AaDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpSlopeR[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpSlopeGr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpSlopeGb[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpSlopeB[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLumaThrR[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLumaThrGr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLumaThrGb[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLumaThrB[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLowOffsetR[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLowOffsetGr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLowOffsetGb[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpLowOffsetB[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpHighOffsetR[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpHighOffsetGr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpHighOffsetGb[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 NpHighOffsetB[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_NR_FILTER_AUTO_ATTR_S;
```

【Member】

Member	Description
LumaStr[8]	The reference image brightness adjusts the denoising strength respectively from dark to bright. The larger the value, the stronger the bright noise removal. Value range： [0x0, 0x1f] Data type： CVI_U8
VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged. Value range： [0x0, 0xff] Data type： CVI_U8
CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region. Value range： [0x0, 0x100] Data type： CVI_U16
CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region. Value range： [0x0, 0x100] Data type： CVI_U16
NonDirFiltStr	Adjusts the strength of noise reduction in the low frequency region. The larger the value, the more noise is removed in the low frequency area. Value range： [0x0, 0x1f] Data type： CVI_U8
VhDirFiltStr	Adjusts the strength of denoising in the horizontal and vertical areas. Larger values remove more noise on horizontal and vertical edges. Value range： [0x0, 0x1f] Data type： CVI_U8
AaDirFiltStr	Adjusts the strength of denoising on diagonal edges. The larger the value, the more noise is removed on the diagonal edges. Value range： [0x0, 0x1f] Data type： CVI_U8
NpSlopeR	The slope of the Noise profile in the R channel Value range： [0x0, 0x3ff] Data type： CVI_U8
NpSlopeGr	The slope of the Noise profile in the Gr channel Value range： [0x0, 0x3ff] Data type： CVI_U8
NpSlopeGb	Noise profile slope in Gb channel Value range： [0x0, 0x3ff] Data type： CVI_U8
NpSlopeB	The slope of the Noise profile on the B channel
	Value range： [0x0, 0x3ff] Data type： CVI_U8
NpLumaThrR	Luminance threshold of the Noise profile in

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRFilterAttr*
- *CVI_ISP_GetNRFilterAttr*

13.3.6 ISP_NR_FILTER_ATTR_S

【Description】

- Bayer noise reduction filter attribute

【Syntax】

```
typedef struct _ISP_NR_FILTER_ATTR_S {
  CVI_U8 TuningMode;
  ISP_NR_FILTER_MANUAL_ATTR_S stManual;
  ISP_NR_FILTER_AUTO_ATTR_S stAuto;
} ISP_NR_FILTER_ATTR_S;
```

【Member】

Member	Description
TuningMode	Output debug strategy, auxiliary adjustment parameters. 8: BNR image result. 11:Flat/Edge Detection Image Result. 12:Vertical Edge Detection Image Result. 13:Horizontal edge detection image result. 14:Anti-diagonal edge detection image result. 15:Diagonal edge detection image result. Value range：[0x0, 0xf] Data type：CVI_U8
stManual	Automatic mode properties.
stAuto	Manual mode properties.

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetNRFilterAttr*
- *CVI_ISP_GetNRFilterAttr*

13.3.7 ISP_RLSC_MANUAL_ATTR_S

【Description】

Bayer Noise Reduction RLSC Manual Properties

【Syntax】

```
typedef struct cviISP_RLSC_MANUAL_ATTR_S {
  CVI_U16 RadialStr;
} ISP_RLSC_MANUAL_ATTR_S;
```

【Member】

Member	Description
RadialStr	Used to adjust RLSC calibrated strength.The larger the value, the closer to the calibrated strength. Value range：[0x0, 0xfff] Data type：CVI_U16

【Note】

None.

【Related Data Type and Interface】

- CVI_ISP_SetRLSCAttr
- CVI_ISP_GetRLSCAttr

13.3.8 ISP_RLSC_AUTO_ATTR_S

【Description】

Bayer noise reduction RLSC automatic attribute

【Syntax】

```
typedef struct cviISP_RLSC_AUTO_ATTR_S {
  CVI_U16 RadialStr[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_RLSC_AUTO_ATTR_S;
```

【Member】

Member	Description
RadialStr	Used to adjust RLSC calibrated strength.The larger the value, the closer to the calibrated strength. Value range：[0x0, 0xfff] Data type：CVI_U16

【Note】

None.

【Related Data Type and Interface】

- CVI_ISP_SetRLSCAttr
- CVI_ISP_GetRLSCAttr

13.3.9 ISP_RLSC_ATTR_S

【Description】

Bayer noise reduction RLSC attribute

【Syntax】

```
typedef struct _ISP_RLSC_ATTR_S {
  CVI_BOOL RlscEnable;
  CVI_U16 RlscCenterX;
  CVI_U16 RlscCenterY;
  ISP_RLSC_MANUAL_ATTR_S stManual;
  ISP_RLSC_AUTO_ATTR_S stAuto;
} ISP_RLSC_ATTR_S;
```

【Member】

Member	Description
RlscEnable	Used to adjust RLSC calibrated strength.The larger the value, the closer to the calibrated strength. Value range: [0, 1] Data type: CVI_BOOL
RlscCenterX	X Center Location Value range: [0x0, 0xfff] Data type: CVI_U16
RlscCenterY	Y Center Location Value range: [0x0, 0x7ff] Data type: CVI_U16
stManual	Bayer Noise Reduction RLSC Manual Mode Properties
stAuto	Bayer Noise Reduction RLSC Automatic Mode Properties

【Note】

None.

【Related Data Type and Interface】

- CVI_ISP_SetRLSCAttr
- CVI_ISP_GetRLSCAttr

14 YNR

14.1 Function Overview

Spatial denoising algorithm for brightness noise in YUV Domain.

14.2 API Referecne

- *CVI_ISP_SetYNRAttr*: Set brightness noise reduction attribute parameters
- *CVI_ISP_GetYNRAttr*: Get brightness noise reduction attribute parameters
- *CVI_ISP_SetYNRMotionNRAttr*: Set the brightness noise reduction moving object attribute parameters
- *CVI_ISP_GetYNRMotionNRAttr*: Get the brightness noise reduction moving object attribute parameters
- *CVI_ISP_SetYNRFilterAttr*: Set brightness noise reduction filter attribute parameters
- *CVI_ISP_GetYNRFilterAttr*: Get brightness noise reduction filter attribute parameters

14.2.1 CVI_ISP_SetYNRAttr

【Description】

Set brightness noise reduction attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetYNRAttr(VI_PIPE ViPipe, const ISP_YNR_ATTR_S *pstYNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRAttr	Brightness Noise Reduction Attribute Parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetYNRAttr*

14.2.2 CVI_ISP_GetYNRAttr

【Description】

Get brightness noise reduction attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetYNRAttr(VI_PIPE ViPipe, ISP_YNR_ATTR_S *pstYNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRAttr	Brightness Noise Reduction Attribute Parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetYNRAttr*

14.2.3 CVI_ISP_SetYNRMotionNRAttr

【Description】

Set the brightness noise reduction moving object attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetYNRMotionNRAttr(VI_PIPE ViPipe, const ISP_YNR_MOTION_NR_ATTR_S *pstYNRMotionNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRMotionNRAttr	Set the brightness noise reduction moving object attribute parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetYNRMotionNRAttr*

14.2.4 CVI_ISP_GetYNRMotionNRAttr

【Description】

Get the brightness noise reduction moving object attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetYNRMotionNRAttr(VI_PIPE ViPipe, ISP_YNR_MOTION_NR_ATTR_S␣
↪*pstYNRMotionNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRMotion-NRAttr	Set the brightness noise reduction moving object attribute parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetYNRMotionNRAttr*

14.2.5 CVI_ISP_SetYNRFilterAttr

【Description】

Set brightness Noise Reduction Filter Attribute Parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetYNRFilterAttr(VI_PIPE ViPipe, const ISP_YNR_FILTER_ATTR_S␣
↪*pstYNRFilterAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRFilterAttr	Brightness Noise Reduction Filter Attribute Parameters	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetYNRFilterAttr*

14.2.6 CVI_ISP_GetYNRFilterAttr

【Description】

Obtaining brightness noise reduction filter attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetYNRFilterAttr(VI_PIPE ViPipe, ISP_YNR_FILTER_ATTR_S
*pstYNRFilterAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYNRFilterAttr	Brightness Noise Reduction Filter Attribute Parameters	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetYNRFilterAttr*

14.3 Data Types

- *ISP_YNR_MANUAL_ATTR_S*: Brightness Noise Reduction Attribute Manual Parameter
- *ISP_YNR_AUTO_ATTR_S*: Brightness Noise Reduction Attribute Automatic Parameters
- *ISP_YNR_ATTR_S*: Brightness Noise Reduction Attribute Parameters
- *ISP_YNR_MOTION_NR_MANUAL_ATTR_S*: Brightness Noise Reduction Moving Object Properties Manual Parameters
- *ISP_YNR_MOTION_NR_AUTO_ATTR_S*: Brightness noise reduction moving object attribute automatic parameters
- *ISP_YNR_MOTION_NR_ATTR_S*: Brightness Noise Reduction Moving Object Attribute Parameters
- *ISP_YNR_FILTER_MANUAL_ATTR_S*: Brightness Noise Reduction Filter Attribute Manual Parameters
- *ISP_YNR_FILTER_AUTO_ATTR_S*: Brightness Noise Reduction Filter Attribute Automatic Parameters
- *ISP_YNR_FILTER_ATTR_S*: Brightness Noise Reduction Filter Attribute Parameters

14.3.1 ISP_YNR_MANUAL_ATTR_S

【Description】

Brightness Noise Reduction Attribute Manual Parameter

【Syntax】

```
typedef struct _ISP_YNR_MANUAL_ATTR_S {
  CVI_U8 WindowType;
  CVI_U8 DetailSmoothMode;
  CVI_U8 NoiseSuppressStr;
  CVI_U8 FilterType;
  CVI_U8 NoiseCoringMax;
  CVI_U8 NoiseCoringBase;
  CVI_U8 NoiseCoringAdv;
} ISP_YNR_MANUAL_ATTR_S;
```

【Member】

Member	Description
WindowType	Denoising filtering locality degree. The smaller the value, the more localized the effect. Value range: [0x0, 0xb] Data type: CVI_U8
DetailSmoothMode	The denoising detail smoothing function is enabled. 0: off. 1: Enabled. Value range: [0x0, 0x1] Data type: CVI_U8
NoiseSuppressStr	Noise suppression strength. The larger the value, the stronger the bright noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
FilterType	Denoising filter strength. The larger the value, the stronger the bright noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringMax	The maximum allowable noise suppression strength. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringBase	The brightness noise tolerance value of the motion area, the judgment of the motion area is linked with the detection of the TNR motion area. The larger the value, the greater the denoising strength of the motion area. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringAdv	Luminance noise tolerance value in the static zone, the judgment of the static zone is linked with the detection of the TNR motion zone. The larger the value, the stronger the noise removal in the static area. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRAttr*

- *CVI_ISP_GetYNRAttr*

14.3.2 ISP_YNR_AUTO_ATTR_S

【Description】

Brightness Noise Reduction Attribute Automatic Parameters

【Syntax】

```
typedef struct _ISP_YNR_AUTO_ATTR_S {
  CVI_U8 WindowType[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DetailSmoothMode[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseSuppressStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 FilterType[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseCoringMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseCoringBase[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseCoringAdv[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_YNR_AUTO_ATTR_S;
```

【Member】

Member	Description
WindowType	Denoising filtering locality degree. The smaller the value, the more localized the effect. Value range: [0x0, 0xb] Data type: CVI_U8
DetailSmoothMode	The denoising detail smoothing function is enabled. 0: off. 1: Enabled. Value range: [0x0, 0x1] Data type: CVI_U8
NoiseSuppressStr	Noise suppression strength. The larger the value, the stronger the bright noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
FilterType	Denoising filter strength. The larger the value, the stronger the bright noise removal. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringMax	The maximum allowable noise suppression strength. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringBase	The brightness noise tolerance value of the motion area, the judgment of the motion area is linked with the detection of the TNR motion area. The larger the value, the greater the denoising strength of the motion area. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseCoringAdv	Luminance noise tolerance value in the static zone, the judgment of the static zone is linked with the detection of the TNR motion zone. The larger the value, the stronger the noise removal in the static area. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRAttr*
- *CVI_ISP_GetYNRAttr*

14.3.3 ISP_YNR_ATTR_S

【Description】

Brightness Noise Reduction Attribute Parameters

【Syntax】

```
typedef struct _ISP_YNR_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_BOOL CoringParamEnable;
  CVI_BOOL FiltModeEnable;
  CVI_U16 FiltMode;
  CVI_U8 TuningMode;
  ISP_YNR_MANUAL_ATTR_S stManual;
  ISP_YNR_AUTO_ATTR_S stAuto;
} ISP_YNR_ATTR_S;
```

【Member】

Member	Description
Enable	The YNR module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
enOpType	Job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
CoringParamEnable	Control whether to use manual coring, if it is 0 NoiseCoringBaseLuma[6] / NoiseCoringBaseOffset[6] / NoiseCoringAdvLuma[6] / NoiseCoringAdvOffset[6] Useless. Value range: [0, 1] Data Type: CVI_BOOL
FiltModeEnable	Filter manual mixing mode enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
FiltMode	Filters with hand-tuned blending weights. Value range: [0, 0x100] Data Type: CVI_U16
TuningMode	Output debug strategy to assist in adjusting parameters. 8: YNR image result. 11: Flat/edge detected image result. (Remove 12, 13, 14, 15) 12: Vertical edge detection image result. 13: Horizontal edge detection image result. 14: Anti-diagonal edge detection image results. 15: Diagonal edge detection image result. Value range: [0x0, 0xf] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRAttr*
- *CVI_ISP_GetYNRAttr*

14.3.4 ISP_YNR_MOTION_NR_MANUAL_ATTR_S

【Description】

Brightness Noise Reduction Moving Object Properties Manual Parameters

【Syntax】

```
typedef struct _ISP_YNR_MOTION_NR_MANUAL_ATTR_S {
  CVI_U8 MotionCoringWgtMax;
  CVI_U16 MotionYnrLut[16];
  CVI_U16 MotionCoringWgt[16];
} ISP_YNR_MOTION_NR_MANUAL_ATTR_S;
```

【Member】

Member	Description
MotionCoringWgtMax	For the object movement area, the maximum value of the noise can be allowed to be preserved. Value range: [0x0, 0xff] Data type: CVI_U8
MotionYnrLut[16]	Adjust the intensity of debrightening and denoising corresponding to the amount of motion of different objects, and divide the amount of motion into 16 levels. The larger the value, the stronger the debrightening strength. Value range: [0x0, 0xff] Data Type: CVI_U16
MotionCoringWgt [16]	Adjust the degree of noise retention corresponding to the amount of motion of different objects, and the amount of motion is divided into 16 levels. The larger the value, the more noise is preserved. Value range: [0x0, 0x100] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRMotionNRAttr*
- *CVI_ISP_GetYNRMotionNRAttr*

14.3.5 ISP_YNR_MOTION_NR_AUTO_ATTR_S

【Description】

Brightness noise reduction moving object attribute automatic parameters

【Syntax】

```
typedef struct _ISP_YNR_MOTION_NR_AUTO_ATTR_S {
  CVI_U8 MotionCoringWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 MotionYnrLut[16][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 MotionCoringWgt[16][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_YNR_MOTION_NR_AUTO_ATTR_S;
```

【Member】

Member	Description
MotionCoringWgtMax	For the object movement area, the maximum value of the noise can be allowed to be preserved. Value range: [0x0, 0xff] Data type: CVI_U8
MotionYnrLut[16]	Adjust the intensity of debrightening and denoising corresponding to the amount of motion of different objects, and divide the amount of motion into 16 levels. The larger the value, the stronger the debrightening strength. Value range: [0x0, 0xff] Data Type: CVI_U16
MotionCoringWgt [16]	Adjust the degree of noise retention corresponding to the amount of motion of different objects, and the amount of motion is divided into 16 levels. The larger the value, the more noise is preserved. Value range: [0x0, 0x100] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRMotionNRAttr*
- *CVI_ISP_GetYNRMotionNRAttr*

14.3.6 ISP_YNR_MOTION_NR_ATTR_S

【Description】

Brightness Noise Reduction Moving Object Attribute Parameters

【Syntax】

```
typedef struct _ISP_YNR_MOTION_NR_ATTR_S {
  ISP_YNR_MOTION_NR_MANUAL_ATTR_S stManual;
  ISP_YNR_MOTION_NR_AUTO_ATTR_S stAuto;
} ISP_YNR_MOTION_NR_ATTR_S;
```

【Member】

Member	Description
stManual	Manual Mode Parameter Properties
stAuto	Automatic Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRMotionNRAttr*
- *CVI_ISP_GetYNRMotionNRAttr*

14.3.7 ISP_YNR_FILTER_MANUAL_ATTR_S

【Description】

Brightness Noise Reduction Filter Attribute Manual Parameters

【Syntax】

```
typedef struct _ISP_YNR_FILTER_MANUAL_ATTR_S {
  CVI_U8 VarThr;
  CVI_U16 CoringWgtLF;
  CVI_U16 CoringWgtHF;
  CVI_U8 NonDirFiltStr;
  CVI_U8 VhDirFiltStr;
  CVI_U8 AaDirFiltStr;
  CVI_U8 CoringWgtMax;
  CVI_U16 FilterMode;
} ISP_YNR_FILTER_MANUAL_ATTR_S;
```

【Member】

Member	Description
VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged. Value range: [0x0, 0xff] Data type: CVI_U8
CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region. Value range: [0x0, 0x100] Data Type: CVI_U16
CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region. Value range: [0x0, 0x100] Data Type: CVI_U16
NonDirFiltStr	Adjusts the strength of noise reduction in the low frequency region. The larger the value, the more noise is removed in the low frequency area. Value range: [0x0, 0x1f] Data type: CVI_U8
VhDirFiltStr	Adjusts the strength of denoising in the horizontal and vertical areas. Larger values remove more noise on horizontal and vertical edges. Value range: [0x0, 0x1f] Data type: CVI_U8
AaDirFiltStr	Adjusts the strength of denoising on diagonal edges. The larger the value, the more noise is removed on the diagonal edges. Value range: [0x0, 0x1f] Data type: CVI_U8
CoringWgtMax	The maximum value of noise that can be allowed to remain. Value range: [0x0, 0xff] Data type: CVI_U8
FilterMode	Filter mode. Smaller values preserve better noise uniformity in flat areas. Value range: [0x0, 0x3ff] Data Type: CVI_U16

【Note】

None.

【relatecd data type and interface】

- *CVI_ISP_SetYNRFilterAttr*
- *CVI_ISP_GetYNRFilterAttr*

14.3.8 ISP_YNR_FILTER_AUTO_ATTR_S

【Description】

Brightness Noise Reduction Filter Attribute Automatic Parameters

【Syntax】

```
typedef struct _ISP_YNR_FILTER_AUTO_ATTR_S {
  CVI_U8 VarThr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 CoringWgtLF[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 CoringWgtHF[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NonDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 VhDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 AaDirFiltStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 CoringWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 FilterMode[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_YNR_FILTER_AUTO_ATTR_S;
```

【Member】

Member	Description
VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged. Value range: [0x0, 0xff] Data type: CVI_U8
CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region. Value range: [0x0, 0x100] Data Type: CVI_U16
CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region. Value range: [0x0, 0x100] Data Type: CVI_U16
NonDirFiltStr	Adjusts the strength of noise reduction in the low frequency region. The larger the value, the more noise is removed in the low frequency area. Value range: [0x0, 0x1f] Data type: CVI_U8
VhDirFiltStr	Adjusts the strength of denoising in the horizontal and vertical areas. Larger values remove more noise on horizontal and vertical edges. Value range: [0x0, 0x1f] Data type: CVI_U8
AaDirFiltStr	Adjusts the strength of denoising on diagonal edges. The larger the value, the more noise is removed on the diagonal edges. Value range: [0x0, 0x1f] Data type: CVI_U8
CoringWgtMax	The maximum value of noise that can be allowed to remain. Value range: [0x0, 0xff] Data type: CVI_U8
FilterMode	Filter mode. Smaller values preserve better noise uniformity in flat areas. Value range: [0x0, 0x3ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRFilterAttr*
- *CVI_ISP_GetYNRFilterAttr*

14.3.9 ISP_YNR_FILTER_ATTR_S

【Description】

Brightness Noise Reduction Filter Attribute Parameters

【Syntax】

```
typedef struct _ISP_YNR_FILTER_ATTR_S {
  ISP_YNR_FILTER_MANUAL_ATTR_S stManual;
  ISP_YNR_FILTER_AUTO_ATTR_S stAuto;
} ISP_YNR_FILTER_ATTR_S;
```

【Member】

Member	Description
stManual	Manual Mode Parameter Properties
stAuto	Automatic Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYNRFilterAttr*
- *CVI_ISP_GetYNRFilterAttr*

15 CNR

15.1 Function Overview

Spatial denoising algorithm for chrominance noise in YUV Domain.

15.2 API Reference

- *CVI_ISP_SetCNRAttr* : Set chroma noise reduction parameter attribute
- *CVI_ISP_GetCNRAttr* : Get chroma noise reduction parameter attribute
- *CVI_ISP_SetCNRMotionNRAttr* : Set the parameter attribute of chroma noise reduction moving object
- *CVI_ISP_GetCNRMotionNRAttr* : Get the parameter attribute of chroma noise reduction moving object

15.2.1 CVI_ISP_SetCNRAttr

【Description】

Set chroma noise reduction parameter attribute

【Syntax】

```
CVI_S32 CVI_ISP_SetCNRAttr(VI_PIPE ViPipe, const ISP_CNR_ATTR_S *pstCNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCNRAttr	Colorimetric Noise Reduction parameter in Properties	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCNRAttr*

15.2.2 CVI_ISP_GetCNRAttr

【Description】

Get chroma noise reduction parameter attribute

【Syntax】

```
CVI_S32 CVI_ISP_GetCNRAttr(VI_PIPE ViPipe, ISP_CNR_ATTR_S *pstCNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCNRAttr	chroma noise reduction parameter attribute	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCNRAttr*

15.2.3 CVI_ISP_SetCNRMotionNRAttr

【Description】

- Set the parameter attribute of chroma noise reduction moving object

【Syntax】

```
CVI_S32 CVI_ISP_SetCNRMotionNRAttr(VI_PIPE ViPipe, const ISP_CNR_MOTION_NR_ATTR_
↪S *pstCNRMotionNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCNRAttr	Chroma noise reduction moving object parameter properties	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCNRMotionNRAttr*

15.2.4 CVI_ISP_GetCNRMotionNRAttr

【Description】

Get the parameter attribute of chroma noise reduction moving object

【Syntax】

```
CVI_S32 CVI_ISP_GetCNRMotionNRAttr(VI_PIPE ViPipe, const ISP_CNR_MOTION_NR_ATTR_
↪S *pstCNRMotionNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCNRAttr	Chroma noise reduction moving object parameter properties	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCNRMotionNRAttr*

15.3 Data Types

- *ISP_CNR_MANUAL_ATTR_S* : Chroma noise reduction parameter manual attribute
- *ISP_CNR_AUTO_ATTR_S* : Chroma noise reduction parameter automatic attribute
- *ISP_CNR_ATTR_S* : chroma noise reduction parameter attribute
- *ISP_CNR_MOTION_NR_MANUAL_ATTR_S* : chroma noise reduction moving object parameter manual attribute

- *ISP_CNR_MOTION_NR_AUTO_ATTR_S* : chroma noise reduction moving object parameter automatic attribute
- *ISP_CNR_MOTION_NR_ATTR_S* : Chroma noise reduction moving object parameter attribute

15.3.1 ISP_CNR_MANUAL_ATTR_S

【Description】

- Chroma noise reduction parameter manual attribute

【Syntax】

```
typedef struct _ISP_CNR_MANUAL_ATTR_S {
  CVI_U8 CnrStr;
  CVI_U8 NoiseSuppressStr;
  CVI_U8 NoiseSuppressGain;
  CVI_U8 FilterType;
  CVI_U8 MotionNrStr;
  CVI_U8 LumaWgt;
  CVI_U8 DetailSmoothMode;
} ISP_CNR_MANUAL_ATTR_S;
```

【Member】

Member	Description
CnrStr	the intensity of color noise removal; the larger the value is, the stronger the denoising intensity is. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressStr	color noise suppression intensity; the larger the value is, the stronger the color noise removal is. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressGain	intensity gain of color noise suppression; the smaller the value is, the stronger the denoising intensity is. Value range: [0x1, 0x8] Data type: CVI_U8
FilterType	the strength of color noise filter; the larger the value is, the stronger the color noise removal is. Value range: [0x0, 0x1f] Data type: CVI_U8
MotionNrStr	Adjust the intensity of color noise removal in the moving area; the larger the value is, the less color noise is in the motion region. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgt	the color denoising weight of reference brightness; the larger the value is, the stronger the intensity of color noise is affected by brightness. Value range: [0x0, 0x8] Data type: CVI_U8
DetailSmoothMode	Enable the de-noising detail smoothing function; 0: close. 1: Enable. Value range: [0x0, 0x1] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRAttr*
- *CVI_ISP_GetCNRAttr*

15.3.2 ISP_CNR_AUTO_ATTR_S

【Description】

Chroma noise reduction parameter automatic attribute

【Syntax】

```
typedef struct _ISP_CNR_AUTO_ATTR_S {
  CVI_U8 CnrStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseSuppressStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 NoiseSuppressGain[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 FilterType[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 MotionNrStr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LumaWgt[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DetailSmoothMode[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CNR_AUTO_ATTR_S;
```

【Member】

Member	Description
CnrStr	the intensity of color noise removal; the larger the value is, the stronger the denoising intensity is. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressStr	color noise suppression intensity; the larger the value is, the stronger the color noise removal is. Value range: [0x0, 0xff] Data type: CVI_U8
NoiseSuppressGain	intensity gain of color noise suppression; the smaller the value is, the stronger the denoising intensity is. Value range: [0x1, 0x8] Data type: CVI_U8
FilterType	the strength of color noise filter; the larger the value is, the stronger the color noise removal is. Value range: [0x0, 0x1f] Data type: CVI_U8
MotionNrStr	Adjust the intensity of color noise removal in the moving area; the larger the value is, the less color noise is in the motion region. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgt	the color denoising weight of reference brightness; the larger the value is, the stronger the intensity of color noise is affected by brightness. Value range: [0x0, 0x8] Data type: CVI_U8
DetailSmoothMode	Enable the de-noising detail smoothing function; 0: close. 1: Enable. Value range: [0x0, 0x1] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRAttr*
- *CVI_ISP_GetCNRAttr*

15.3.3 ISP_CNR_ATTR_S

【Description】

Chroma noise reduction parameter attribute.

【Syntax】

```
typedef struct _ISP_CNR_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  ISP_CNR_MANUAL_ATTR_S stManual;
  ISP_CNR_AUTO_ATTR_S stAuto;
} ISP_CNR_ATTR_S;
```

【Member】

Member	Description
Enable	Enable CNR module; 0: close. 1: enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	Type of work OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Manual mode parameter properties
stAuto	Automatic mode parameter properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRAttr*
- *CVI_ISP_GetCNRAttr*

15.3.4 ISP_CNR_MOTION_NR_MANUAL_ATTR_S

【Description】

Chroma Noise Reduction Moving Object parameters Manual Properties

【Syntax】

```
typedef struct _ISP_CNR_MOTION_NR_MANUAL_ATTR_S {
  CVI_U8 MotionCnrCoringLut[16]
  CVI_U8 MotionCnrStrLut[16];
} ISP_CNR_MOTION_NR_MANUAL_ATTR_S;
```

【Member】

Member	Description
MotionCnrCoringLut[16]	Use LUT to adjust the intensity of color noise suppression corresponding to the amount of motion of different objects, and the amount of motion is divided into 16 levels. The larger the value, the stronger the denoising strength Value range: [0x0, 0xff] Data type: CVI_U8
MotionCnrStrLut[16]	Use LUT to adjust the denoising intensity corresponding to different object movement, and divide the movement into 16 levels. The larger the value, the stronger the denoising strength. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRMotionNRAttr*
- *CVI_ISP_GetCNRMotionNRAttr*

15.3.5 ISP_CNR_MOTION_NR_AUTO_ATTR_S

【Description】

Chroma Noise Reduction Moving Object parameters Automatic Properties

【Syntax】

```
typedef struct _ISP_CNR_MOTION_NR_ATTR_S {
  CVI_BOOL MotionCnrEnable;
  ISP_CNR_MOTION_NR_MANUAL_ATTR_S stManual;
  ISP_CNR_MOTION_NR_AUTO_ATTR_S stAuto;
} ISP_CNR_MOTION_NR_ATTR_S;
```

【Member】

Member	Description
MotionCnrEnable	The function of adjusting the strength of denoising by referring to the amount of motion of the object is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
stManual	Manual Mode parameter Properties
stAuto	Auto Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRMotionNRAttr*
- *CVI_ISP_GetCNRMotionNRAttr*

15.3.6 ISP_CNR_MOTION_NR_ATTR_S

【Description】

Chroma Noise Reduction Moving Object parameter Properties

【Syntax】

```
typedef struct _ISP_CNR_MOTION_NR_ATTR_S {
  CVI_BOOL MotionCnrEnable;
  ISP_CNR_MOTION_NR_MANUAL_ATTR_S stManual;
  ISP_CNR_MOTION_NR_AUTO_ATTR_S stAuto;
} ISP_CNR_MOTION_NR_ATTR_S;
```

【Member】

Member	Description
MotionCnrEnable	The function of adjusting the strength of denoising by referring to the amount of motion of the object is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
stManual	Manual Mode parameter Properties
stAuto	Auto Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCNRMotionNRAttr*
- *CVI_ISP_GetCNRMotionNRAttr*

16 TNR

16.1 Function Overview

The time domain denoising algorithm in YUV domain is also called 3DNR.

16.2 API Reference

- *CVI_ISP_SetTNRAttr*: Set TNR property parameter.
- *CVI_ISP_GetTNRAttr*: Get TNR property parameter.
- *CVI_ISP_SetTNRNoiseModelAttr*: Set TNR Noise Model property parameter.
- *CVI_ISP_GetTNRNoiseModelAttr*: Get TNR Noise Model property parameter.
- *CVI_ISP_SetTNRLumaMotionAttr*: Set TNR brightness to intensity gain characteristic table.
- *CVI_ISP_GetTNRLumaMotionAttr*: Get TNR brightness to intensity gain characteristic table.
- *CVI_ISP_SetTNRGhostAttr*: Set TNR momentum to tail elimination degree characteristic table.
- *CVI_ISP_GetTNRGhostAttr*: Get TNR momentum to tail elimination degree characteristic table.
- *CVI_ISP_SetTNRMtPrtAttr*: Set TNR momentum protection attribute parameter.
- *CVI_ISP_GetTNRMtPrtAttr*: Get TNR momentum protection attribute parameter.
- *CVI_ISP_SetTNRMotionAdaptAttr*: Set TNR momentum to strength gain characteristic table
- *CVI_ISP_GetTNRMotionAdaptAttr*: Get TNR momentum to strength gain characteristic table

16.2.1 CVI_ISP_SetTNRAttr

【Description】

Set TNR property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRAttr(VI_PIPE ViPipe, const ISP_TNR_ATTR_S *pstTNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRAttr	TNR property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

```
VI_PIPE ViPipe = 0;
ISP_TNR_ATTR_S stAttr;
CVI_ISP_GetTNRAttr(ViPipe, &stAttr);
stAttr.enOpType = OP_TYPE_AUTO;
CVI_ISP_SetTNRAttr(ViPipe, &stAttr);
```

【Related Topic】

- *CVI_ISP_GetTNRAttr*

16.2.2 CVI_ISP_GetTNRAttr

【Description】

Get TNR property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetTNRAttr(VI_PIPE ViPipe, ISP_TNR_ATTR_S *pstTNRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRAttr	TNR property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRAttr*

16.2.3 CVI_ISP_SetTNRNoiseModelAttr

【Description】

Set TNR Noise Model property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRNoiseModelAttr(VI_PIPE ViPipe, const ISP_TNR_NOISE_MODEL_
↪ATTR_S *pstTNRNoiseModelAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRNoiseModelAttr	TNR Noise Model property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetTNRNoiseModelAttr*

16.2.4 CVI_ISP_GetTNRNoiseModelAttr

【Description】

Get TNR Noise Model property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetTNRNoiseModelAttr(VI_PIPE ViPipe, ISP_TNR_NOISE_MODEL_ATTR_S *pstTNRNoiseModelAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRNoiseModelAttr	TNR Noise Model property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRNoiseModelAttr*

16.2.5 CVI_ISP_SetTNRLumaMotionAttr

【Description】

Set TNR brightness to intensity gain characteristic table.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRLumaMotionAttr(VI_PIPE ViPipe, const ISP_TNR_LUMA_MOTION_
↪ATTR_S *pstTNRLumaMotionAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRLumaMotionAttr	TNR brightness to intensity gain characteristic table	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetTNRLumaMotionAttr*

16.2.6 CVI_ISP_GetTNRLumaMotionAttr

【Description】

Get TNR brightness to intensity gain characteristic table.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRLumaMotionAttr(VI_PIPE ViPipe, const ISP_TNR_LUMA_MOTION_
↪ATTR_S *pstTNRLumaMotionAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRLumaMotionAttr	TNR brightness to intensity gain characteristic table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRLumaMotionAttr*

16.2.7 CVI_ISP_SetTNRGhostAttr

【Description】

Set TNR momentum to tail elimination degree characteristic table.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRGhostAttr(VI_PIPE ViPipe, const ISP_TNR_GHOST_ATTR_S␣
↪*pstTNRGhostAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRGhostAttr	TNR momentum to tail elimination degree characteristic table	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetTNRGhostAttr*

16.2.8 CVI_ISP_GetTNRGhostAttr

【Description】

Get TNR momentum to tail elimination degree characteristic table.

【Syntax】

```
CVI_S32 CVI_ISP_GetTNRGhostAttr(VI_PIPE ViPipe, ISP_TNR_GHOST_ATTR_S␣
↪*pstTNRGhostAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRGhostAttr	TNR momentum to tail elimination degree characteristic table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRGhostAttr*

16.2.9 CVI_ISP_SetTNRMtPrtAttr

【Description】

Set TNR momentum protection attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRMtPrtAttr(VI_PIPE ViPipe, const ISP_TNR_MT_PRT_ATTR_S
↪*pstTNRMtPrtAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRMtPrtAttr	TNR momentum protection attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetTNRMtPrtAttr*

16.2.10 CVI_ISP_GetTNRMtPrtAttr

【Description】

Get TNR momentum protection attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetTNRMtPrtAttr(VI_PIPE ViPipe, ISP_TNR_MT_PRT_ATTR_S␣
↪*pstTNRMtPrtAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRMtPrtAttr	TNR momentum protection attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRMtPrtAttr*

16.2.11 CVI_ISP_SetTNRMotionAdaptAttr

【Description】

- Set TNR momentum to strength gain characteristic table

【Syntax】

```
CVI_S32 CVI_ISP_SetTNRMotionAdaptAttr(VI_PIPE ViPipe, const ISP_TNR_MOTION_
↪ADAPT_ATTR_S *pstTNRMotionAdaptAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRMotionAdaptAttr	TNR momentum to strength gain characteristic table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetTNRMotionAdaptAttr*

16.2.12 CVI_ISP_GetTNRMotionAdaptAttr

【Description】

- Get TNR momentum to strength gain characteristic table

【Syntax】

```
CVI_S32 CVI_ISP_GetTNRMotionAdaptAttr(VI_PIPE ViPipe, ISP_TNR_MOTION_ADAPT_ATTR_
↪S *pstTNRMotionAdaptAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstTNRMotionAdaptAttr	TNR momentum to strength gain characteristic table	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetTNRMotionAdaptAttr*

16.3 Data Types

- *ISP_TNR_MANUAL_ATTR_S*: TNR attribute manual parameter.
- *ISP_TNR_AUTO_ATTR_S*: TNR attribute auto parameter.
- *ISP_TNR_ATTR_S*: TNR attribute parameter.
- *ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S*: TNR Noise Model attribute manual parameter.

- *ISP_TNR_NOISE_MODEL_AUTO_ATTR_S*: TNR Noise Model attribute auto parameter.
- *ISP_TNR_NOISE_MODEL_ATTR_S*: TNR Noise Model attribute parameter.
- *ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S*: Characteristic table of brightness to intensity gain of TNR in manual mode.
- *ISP_TNR_LUMA_MOTION_AUTO_ATTR_S*: Characteristic table of brightness to intensity gain of TNR in auto mode.
- *ISP_TNR_LUMA_MOTION_ATTR_S*: TNR brightness to intensity gain characteristic table.
- *ISP_TNR_GHOST_MANUAL_ATTR_S*: Characteristic table of momentum to tailing elimination degree in manual mode.
- *ISP_TNR_GHOST_AUTO_ATTR_S*: Characteristic table of momentum to tailing elimination degree in auto mode.
- *ISP_TNR_GHOST_ATTR_S*: Characteristic table of momentum to tailing elimination degree.
- *ISP_TNR_MT_PRT_MANUAL_ATTR_S*: Manual mode TNR momentum protection attribute parameter.
- *ISP_TNR_MT_PRT_AUTO_ATTR_S*: Auto mode TNR momentum protection attribute parameter.
- *ISP_TNR_MT_PRT_ATTR_S*: TNR momentum protection attribute parameter.
- *ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S*: Parameter Description of momentum vs. strength gain in manual mode
- *ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S*: Parameter Description for momentum vs. strength gain in automatic mode
- *ISP_TNR_MOTION_ADAPT_ATTR_S*: Parameter Description of momentum versus strength gain

16.3.1 ISP_TNR_MANUAL_ATTR_S

【Description】

TNR attribute manual parameter.

【Syntax】

```
typedef struct _ISP_TNR_MANUAL_ATTR_S {
  CVI_U8 TnrStrength0;
  CVI_U8 MapThdLow0;
  CVI_U8 MapThdHigh0;
  CVI_U8 MtDetectUnit;
  CVI_S16 BrightnessNoiseLevelLE;
  CVI_S16 BrightnessNoiseLevelSE;
  CVI_BOOL MtFiltMode;
```

(continues on next page)

(continued from previous page)

```
  CVI_U16 MtFiltWgt;
} ISP_TNR_MANUAL_ATTR_S;
```

【Member】

Member	Description
TnrStrength0	Long Exposure TNR Intensity Gain Value range: [0x0, 0xff] Data type: CVI_U8
MapThdLow0	Long Exposure TNR Intensity Upper Limit Value range: [0x0, 0xff] Data type: CVI_U8
MapThdHigh0	Long exposure TNR intensity lower limit Value range: [0x0, 0xff] Data type: CVI_U8
MtDetectUnit	The anti-noise ability of motion detection, the larger the value, the stronger the anti-noise ability, but the lower the fineness of detection. Value range: [0x3, 0x6] Data type: CVI_U8
BrightnessNoiseLevelLE	Long Exposure Luminance Noise Tolerance Value Value range: [0x1, 0x3ff] Data type: CVI_S16
BrightnessNoiseLevelSE	Short exposure luminance noise tolerance value Value range: [0x1, 0x3ff] Data type: CVI_S16
MtFiltMode	Motion Detection Filter Mode Value range: [0x00, 0x01] Data type: CVI_BOOL
MtFiltWgt	Motion Detection Filter Weights Value range: [0x1, 0x100] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRAttr*
- *CVI_ISP_GetTNRAttr*

16.3.2 ISP_TNR_AUTO_ATTR_S

【Description】

TNR attribute auto parameter.

【Syntax】

```
typedef struct _ISP_TNR_AUTO_ATTR_S {
  CVI_U8 TnrStrength0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 MapThdLow0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 MapThdHigh0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 MtDetectUnit[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_S16 BrightnessNoiseLevelLE[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_S16 BrightnessNoiseLevelSE[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_BOOL MtFiltMode[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 MtFiltWgt[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_AUTO_ATTR_S;
```

【Member】

Member	Description
TnrStrength0	Long Exposure TNR Intensity Gain Value range: [0x0, 0xff] Data type: CVI_U8
MapThdLow0	Long Exposure TNR Intensity Upper Limit Value range: [0x0, 0xff] Data type: CVI_U8
MapThdHigh0	Long exposure TNR intensity lower limit Value range: [0x0, 0xff] Data type: CVI_U8
MtDetectUnit	The anti-noise ability of motion detection, the larger the value, the stronger the anti-noise ability, but the lower the fineness of detection. Value range: [0x3, 0x6] Data type: CVI_U8
BrightnessNoiseLevelLE	Long Exposure Luminance Noise Tolerance Value Value range: [0x1, 0x3ff] Data type: CVI_S16
BrightnessNoiseLevelSE	Short exposure luminance noise tolerance value Value range: [0x1, 0x3ff] Data type: CVI_S16
MtFiltMode	Motion Detection Filter Mode Value range: [0x00, 0x01] Data type: CVI_BOOL
MtFiltWgt	Motion Detection Filter Weights Value range: [0x1, 0x100] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRAttr*
- *CVI_ISP_GetTNRAttr*

16.3.3 ISP_TNR_ATTR_S

【Description】

TNR attribute parameter.

【Syntax】

```
typedef struct _ISP_TNR_ATTR_S {
  CVI_BOOL Enable;
  ISP_

     OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_BOOL TuningMode;
  CVI_BOOL TnrMtMode;
  CVI_BOOL YnrCnrSharpenMtMode;
  CVI_BOOL PreSharpenMtMode;
  CVI_U8 ChromaScalingDownMode;
  CVI_BOOL CompGainEnable;
  ISP_TNR_MANUAL_ATTR_S stManual;
  ISP_TNR_AUTO_ATTR_S stAuto;
} ISP_TNR_ATTR_S;
```

【Member】

Member	Description
Enable	The TNR module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
TuningMode	Debug mode, output visual auxiliary information to help users debug 0: Do not output visual aids 1: Output the visual result of motion detection. The brighter the motion, the more obvious the motion, and the darker it is, the opposite. Value range: [0, 1] Data type: CVI_BOOL
TnrMtMode	TNR Motion mode: 0: Motion IIR (represents that the detected motion has been processed by IIR) 1: Motion history (meaning that the detected motion has not been processed by IIR) Value range: [0x0, 0x1] Data type: CVI_BOOL
YnrCnrSharpenMtMode	YNR/CNR Motion mode: 0: Motion IIR (represents that the detected motion has been processed by IIR) 1: Motion history (meaning that the detected motion has not been processed by IIR) Value range: [0x0, 0x1] Data type: CVI_BOOL
PreSharpenMtMode	Presharpen Motion mode: 0: Motion IIR (represents that the detected motion has been processed by IIR) 1: Motion history (meaning that the detected motion has not been processed by IIR) Value range: [0x0, 0x1] Data type: CVI_BOOL
ChromaScalingDownMode	Chroma downscaling mode
CompGainEnable	Brightness compensation function enable Value range: [0, 1] Data type: CVI_BOOL
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRAttr*
- *CVI_ISP_GetTNRAttr*

16.3.4 ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S

【Description】

TNR Noise Model attribute auto parameter.

【Syntax】

```
typedef struct _ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S {
  CVI_U8 RNoiseLevel0;
  CVI_U8 GNoiseLevel0;
  CVI_U8 BNoiseLevel0;
  CVI_U8 RNoiseLevel1;
  CVI_U8 GNoiseLevel1;
  CVI_U8 BNoiseLevel1;
  CVI_U8 RNoiseHiLevel0;
  CVI_U8 GNoiseHiLevel0;
  CVI_U8 BNoiseHiLevel0;
  CVI_U8 RNoiseHiLevel1;
  CVI_U8 GNoiseHiLevel1;
  CVI_U8 BNoiseHiLevel1;
} ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S;
```

【Member】

Member	Description
RNoiseLevel0	Noise tolerance of long exposure red channel; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseHiLevel0	Long exposure red channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseLevel0	Noise tolerance of long exposure green channel; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseHiLevel0	Long exposure green channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseLevel0	Noise tolerance of long exposure blue channel; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseHiLevel0	Long exposure blue channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseLevel1	Short exposure red channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseHiLevel1	Short exposure red channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseLevel1	Short exposure green channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseHiLevel1	Short exposure green channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseLevel1	Short exposure blue channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseHiLevel1	Short exposure blue channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRNoiseModelAttr*
- *CVI_ISP_GetTNRNoiseModelAttr*

16.3.5 ISP_TNR_NOISE_MODEL_AUTO_ATTR_S

【Description】

TNR Noise Model attribute auto parameter.

【Syntax】

```
typedef struct _ISP_TNR_NOISE_MODEL_AUTO_ATTR_S {
  CVI_U8 RNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 GNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BNoiseLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 RNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 GNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BNoiseLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 RNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 GNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BNoiseHiLevel0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 RNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 GNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BNoiseHiLevel1[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_NOISE_MODEL_AUTO_ATTR_S;
```

【Member】

Member	Description
RNoiseLevel0	Noise tolerance of long exposure red channel; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseHiLevel0	Long exposure red channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseLevel0	Noise tolerance of long exposure green channel; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseHiLevel0	Long exposure green channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseLevel0	Noise tolerance of long exposure blue channel; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseHiLevel0	Long exposure blue channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseLevel1	Short exposure red channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
RNoiseHiLevel1	Short exposure red channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseLevel1	Short exposure green channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
GNoiseHiLevel1	Short exposure green channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseLevel1	Short exposure blue channel noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8
BNoiseHiLevel1	Short exposure blue channel brightness noise tolerance; Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRNoiseModelAttr*
- *CVI_ISP_GetTNRNoiseModelAttr*

16.3.6 ISP_TNR_NOISE_MODEL_ATTR_S

【Description】

TNR Noise Model attribute parameter.

【Syntax】

```
typedef struct _ISP_TNR_NOISE_MODEL_ATTR_S {
  ISP_TNR_NOISE_MODEL_MANUAL_ATTR_S stManual;
  ISP_TNR_NOISE_MODEL_AUTO_ATTR_S stAuto;
} ISP_TNR_NOISE_MODEL_ATTR_S;
```

【Member】

Member	Description
stManual	Manual mode parameter properties
stAuto	Automatic mode parameter properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRNoiseModelAttr*
- *CVI_ISP_GetTNRNoiseModelAttr*

16.3.7 ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S

【Description】

Characteristic table of brightness to intensity gain of TNR in manual mode.

【Syntax】

```
typedef struct _ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S {
  CVI_U16 L2mIn0[4];
  CVI_U8 L2mOut0[4];
  CVI_U16 L2mIn1[4];
  CVI_U8 L2mOut1[4];
  CVI_BOOL MtLumaMode;
} ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S;
```

【Member】

Member	Description
L2mIn0[4]	Long exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the gray level; the larger the value, the higher the gray level. Value range: [0x0, 0xfff] Data type: CVI_U16
L2mOut0[4]	Long exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the intensity gain; the greater the value, the stronger the intensity. Value range: [0x0, 0x3f] Data type: CVI_U8
L2mIn1[4]	Short exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the gray level; the larger the value, the higher the gray level. Value range: [0x0, 0xfff] Data type: CVI_U16
L2mOut1[4]	Short exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the intensity gain; the greater the value, the stronger the intensity. Value range: [0x0, 0x3f] Data type: CVI_U8
MtLumaMode	Luma gain reference object 0: luma 1: max(R,G,B) Value range: [0x0, 0x1] Data type: CVI_BOOL

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRLumaMotionAttr*
- *CVI_ISP_GetTNRLumaMotionAttr*

16.3.8 ISP_TNR_LUMA_MOTION_AUTO_ATTR_S

【Description】

Characteristic table of brightness to intensity gain of TNR in auto mode.

【Syntax】

```
typedef struct _ISP_TNR_LUMA_MOTION_AUTO_ATTR_S {
  CVI_U16 L2mIn0[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 L2mOut0[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 L2mIn1[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 L2mOut1[4][ISP_AUTO_ISO_STRENGTH_NUM];
```

(continues on next page)

(continued from previous page)

```
  CVI_BOOL MtLumaMode[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_LUMA_MOTION_AUTO_ATTR_S;
```

【Member】

Member	Description
L2mIn0[4]	Long exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the gray level; the larger the value, the higher the gray level. Value range: [0x0, 0xfff] Data type: CVI_U16
L2mOut0[4]	Long exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the intensity gain; the greater the value, the stronger the intensity. Value range: [0x0, 0x3f] Data type: CVI_U8
L2mIn1[4]	Short exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the gray level; the larger the value, the higher the gray level. Value range: [0x0, 0xfff] Data type: CVI_U16
L2mOut1[4]	Short exposure TNR brightness to intensity gain characteristic table; it consists of an array of four sets of values; define the intensity gain; the greater the value, the stronger the intensity. Value range: [0x0, 0x3f] Data type: CVI_U8
MtLumaMode	Luma gain reference object 0: luma 1: max(R,G,B) Value range: [0x0, 0x1] Data type: CVI_BOOL

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRLumaMotionAttr*
- *CVI_ISP_GetTNRLumaMotionAttr*

16.3.9 ISP_TNR_LUMA_MOTION_ATTR_S

【Description】

TNR brightness to intensity gain characteristic table.

【Syntax】

```
typedef struct _ISP_TNR_LUMA_MOTION_ATTR_S {
  ISP_TNR_LUMA_MOTION_MANUAL_ATTR_S stManual;
  ISP_TNR_LUMA_MOTION_AUTO_ATTR_S stAuto;
} ISP_TNR_LUMA_MOTION_ATTR_S;
```

【Member】

Member	Description
stManual	Manual mode parameter properties
stAuto	Automatic mode parameter properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRLumaMotionAttr*
- *CVI_ISP_GetTNRLumaMotionAttr*

16.3.10 ISP_TNR_GHOST_MANUAL_ATTR_S

【Description】

Characteristic table of momentum to tailing elimination degree in manual mode.

【Syntax】

```
typedef struct _ISP_TNR_GHOST_MANUAL_ATTR_S {
  CVI_U8 PrvMotion0[4];
  CVI_U8 PrtctWgt0[4];
  CVI_U8 MotionHistoryStr;
} ISP_TNR_GHOST_MANUAL_ATTR_S;
```

【Member】

Member	Description
PrvMotion0[4]	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level. Value range: [0x0, 0xff] Data type: CVI_U8
PrtctWgt0[4]	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Defines the intensity gain, the larger the value, the stronger the intensity. Value range: [0x0, 0xf] Data type: CVI_U8
MotionHistoryStr	Defines the degree of retention of the movement track exported to the YNR reference. The higher the value, the longer the trajectory will be retained, otherwise the shorter the trajectory will be retained Value range: [0x0, 0xf] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRGhostAttr*
- *CVI_ISP_GetTNRGhostAttr*

16.3.11 ISP_TNR_GHOST_AUTO_ATTR_S

【Description】

Characteristic table of momentum to tailing elimination degree in auto mode.

【Syntax】

```
typedef struct _ISP_TNR_GHOST_AUTO_ATTR_S {
  CVI_U8 PrvMotion0[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 PrtctWgt0[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 MotionHistoryStr[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_GHOST_AUTO_ATTR_S;
```

【Member】

Member	Description
PrvMotion0[4]	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level. Value range: [0x0, 0xff] Data type: CVI_U8
PrtctWgt0[4]	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Defines the intensity gain, the larger the value, the stronger the intensity. Value range: [0x0, 0xf] Data type: CVI_U8
MotionHistoryStr	Defines the degree of retention of the movement track exported to the YNR reference. The higher the value, the longer the trajectory will be retained, otherwise the shorter the trajectory will be retained Value range: [0x0, 0xf] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRGhostAttr*
- *CVI_ISP_GetTNRGhostAttr*

16.3.12 ISP_TNR_GHOST_ATTR_S

【Description】

Characteristic table of momentum to tailing elimination degree.

【Syntax】

```
typedef struct _ISP_TNR_GHOST_ATTR_S {
  ISP_TNR_GHOST_MANUAL_ATTR_S stManual;
  ISP_TNR_GHOST_AUTO_ATTR_S stAuto;
} ISP_TNR_GHOST_ATTR_S;
```

【Member】

Member	Description
stManual	Manual mode parameter properties
stAuto	Automatic mode parameter properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRGhostAttr*
- *CVI_ISP_GetTNRGhostAttr*

16.3.13 ISP_TNR_MT_PRT_MANUAL_ATTR_S

【Description】

Manual mode TNR momentum protection attribute parameter.

【Syntax】

```
typedef struct _ISP_TNR_MT_PRT_MANUAL_ATTR_S {
  CVI_U8 LowMtPrtLevelY;
  CVI_U8 LowMtPrtLevelU;
  CVI_U8 LowMtPrtLevelV;
  CVI_U8 LowMtPrtInY[4];
  CVI_U8 LowMtPrtInU[4];
  CVI_U8 LowMtPrtInV[4];
  CVI_U8 LowMtPrtOutY[4];
  CVI_U8 LowMtPrtOutU[4];
  CVI_U8 LowMtPrtOutV[4];
  CVI_U8 LowMtPrtAdvIn[4];
  CVI_U8 LowMtPrtAdvOut[4];
} ISP_TNR_MT_PRT_MANUAL_ATTR_S;
```

【Member】

Member	Description
LowMtPrtLevelY	Y channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtLevelU	U channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtLevelV	V channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInY[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of Y channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInU[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of U channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInV[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of V channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutY[4]	Define the vertical axis of the LUT, that is, the protection degree of the front and rear frame difference of the Y channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutU[4]	Define the vertical axis of the LUT, that is, the degree of protection of the frame difference before and after the U channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutV[4]	Define the vertical axis of the LUT, that is, the protection degree of the frame difference before and after the V channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtAdvIn[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of blocks Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtAdvOut[4]	Defines the vertical axis of the LUT, i.e. the protection gain of the difference between the front and back frames Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMtPrtAttr*
- *CVI_ISP_GetTNRMtPrtAttr*

16.3.14 ISP_TNR_MT_PRT_AUTO_ATTR_S

【Description】

Auto mode TNR momentum protection attribute parameter.

【Syntax】

```
typedef struct _ISP_TNR_MT_PRT_AUTO_ATTR_S {
  CVI_U8 LowMtPrtLevelY[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtLevelU[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtLevelV[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtInY[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtInU[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtInV[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtOutY[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtOutU[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtOutV[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtAdvIn[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LowMtPrtAdvOut[4][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_MT_PRT_AUTO_ATTR_S;
```

【Member】

Member	Description
LowMtPrtLevelY	Y channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtLevelU	U channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtLevelV	V channel protection upper limit. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInY[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of Y channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInU[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of U channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtInV[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of V channel pixels. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutY[4]	Define the vertical axis of the LUT, that is, the protection degree of the front and rear frame difference of the Y channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutU[4]	Define the vertical axis of the LUT, that is, the degree of protection of the frame difference before and after the U channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtOutV[4]	Define the vertical axis of the LUT, that is, the protection degree of the frame difference before and after the V channel. Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtAdvIn[4]	Define the horizontal axis of the LUT, that is, the difference between the front and rear frames in units of blocks Value range: [0x0, 0xff] Data type: CVI_U8
LowMtPrtAdvOut[4]	Defines the vertical axis of the LUT, i.e. the protection gain of the difference between the front and back frames Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMtPrtAttr*
- *CVI_ISP_GetTNRMtPrtAttr*

16.3.15 ISP_TNR_MT_PRT_ATTR_S

【Description】

TNR momentum protection attribute parameter.

【Syntax】

```
typedef struct _ISP_TNR_MT_PRT_ATTR_S {
  CVI_BOOL LowMtPrtEn; /*RW; Range:[0, 1]*/
  CVI_BOOL LowMtLowPassEnable; /*RW; Range:[0, 1]*/
  CVI_BOOL LowMtPrtAdvLumaEnable; /*RW; Range:[0, 1]*/
  CVI_BOOL LowMtPrtAdvMode; /*RW; Range:[0, 1]*/
  CVI_U8 LowMtPrtAdvMax; /*RW; Range:[0, 0xff]*/
  CVI_BOOL LowMtPrtAdvDebugMode; /*RW; Range:[0, 1]*/
  CVI_U8 LowMtPrtAdvDebugIn[4]; /*RW; Range:[0x0, 0xff]*/
  CVI_U8 LowMtPrtAdvDebugOut[4]; /*RW; Range:[0x0, 0xff]*/
  ISP_TNR_MT_PRT_MANUAL_ATTR_S stManual;
  ISP_TNR_MT_PRT_AUTO_ATTR_S stAuto;
} ISP_TNR_MT_PRT_ATTR_S;
```

【Member】

Member	Description
LowMtPrtEn	Airspace noise reduction micro motion protection is enabled. This function can define the degree of protection according to the difference between the front and rear frames in units of pixels. The higher the degree of protection, the more inclined to output the original pixel value. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0, 1] Data type: CVI_BOOL
LowMtLowPassEnable	Weak motion detection is low-pass filtered using Value range: [0, 1] Data type: CVI_BOOL
LowMtPrtAdvLumaEnable	The airspace noise reduction micro motion protection function is enabled. This function can define the degree of protection according to the difference between the front and rear frames in units of blocks. The higher the degree of protection, the more inclined the output of the original pixel value. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction Value range: [0, 1] Data type: CVI_BOOL
LowMtPrtAdvMode	Airspace Noise Reduction Micro Motion Mode Value range: [0, 1] Data type: CVI_BOOL
LowMtPrtAdvMax	Protection Gain Cap Value range: [0, 0xff] Data type: CVI_U8
LowMtPrtAdvDebugMode	Micro motion debug mode, output visual auxiliary information to help users debug 0: Output motion detection visualization results. 1: Output motion detection visualization results according to LowMtPrtDebug LUT Value range: [0, 1] Data type: CVI_BOOL
LowMtPrtAdvDebugIn[4]	Debug mode, micro motion input range setting Value range: [0, 0xff] Data type: CVI_U8
LowMtPrtAdvDebugOut[4]	Debug mode, micro motion output value setting Value range: [0, 0xff] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMtPrtAttr*

- *CVI_ISP_GetTNRMtPrtAttr*

16.3.16 ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S

【Description】

- Parameter Description of momentum vs.

  strength gain in manual mode

【Syntax】

```
typedef struct _ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S {
  CVI_U8 AdaptNrLumaStrIn[4];
  CVI_U8 AdaptNrLumaStrOut[4];
  CVI_U8 AdaptNrChromaStrIn[4];
  CVI_U8 AdaptNrChromaStrOut[4];
} ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S;
```

【Member】

Member	Description
AdaptNrLumaStrIn	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrLumaStrOut	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrChromaStrIn	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. Define the chromaticity level, the larger the value, the higher the chromaticity. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrChromaStrOut	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMotionAdaptAttr*
- *CVI_ISP_GetTNRMotionAdaptAttr*

16.3.17 ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S

【Description】

Parameter Description for momentum vs. strength gain in automatic mode

【Syntax】

```
typedef struct _ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S {
  CVI_U8 AdaptNrLumaStrIn[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 AdaptNrLumaStrOut[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 AdaptNrChromaStrIn[4][ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 AdaptNrChromaStrOut[4][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S;
```

【Member】

Member	Description
AdaptNrLumaStrIn	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrLumaStrOut	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrChromaStrIn	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. Define the chromaticity level, the larger the value, the higher the chromaticity. Value range: [0x0, 0xff] Data type: CVI_U8
AdaptNrChromaStrOut	TNR Momentum vs Strength Gain Characteristics Table. An array of four sets of values. When the value is higher, the protected motion area will be more inclined to the original pixel value output. Conversely, the lower the value, the more inclined the pixel value output after temporal noise reduction. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMotionAdaptAttr*
- *CVI_ISP_GetTNRMotionAdaptAttr*

16.3.18 ISP_TNR_MOTION_ADAPT_ATTR_S

【Description】

- Parameter Description of momentum versus strength gain

【Syntax】

```
typedef struct _ISP_TNR_MOTION_ADAP_ATTR_S {
  ISP_TNR_MOTION_ADAPT_MANUAL_ATTR_S stManual;
  ISP_TNR_MOTION_ADAPT_AUTO_ATTR_S stAuto;
} ISP_TNR_MOTION_ADAPT_ATTR_S;
```

【Member】

Member	Description
stManual	Manual mode parameter attribute
stAuto	Auto mode parameter attribute

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetTNRMotionAdaptAttr*
- *CVI_ISP_GetTNRMotionAdaptAttr*

17 Crosstalk

17.1 Function Overview

It can balance the difference between adjacent pixels Gr and Gb between raw

and effectively prevent the square

or other similar patterns produced by demosaic interpolation algorithm.

17.2 API Reference

- *CVI_ISP_SetCrosstalkAttr* : Set Crosstalk property parameter.
- *CVI_ISP_GetCrosstalkAttr* : Get Crosstalk property parameter.

17.2.1 CVI_ISP_SetCrosstalkAttr

【Description】

Set Crosstalk property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetCrosstalkAttr(VI_PIPE ViPipe, const ISP_CROSSTALK_ATTR_S
*pstCrosstalkAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCrosstalkAttr	Crosstalk property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCrosstalkAttr*

17.2.2 CVI_ISP_GetCrosstalkAttr

【Description】

Get Crosstalk property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetCrosstalkAttr(VI_PIPE ViPipe, ISP_CROSSTALK_ATTR_S␣
↪*pstCrosstalkAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCrosstalkAttr	Crosstalk property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCrosstalkAttr*

17.3 Data Types

- *ISP_CROSSTALK_MANUAL_ATTR_S* : Crosstalk attribute manual parameter.
- *ISP_CROSSTALK_AUTO_ATTR_S* : Crosstalk attribute auto parameter.
- *ISP_CROSSTALK_ATTR_S* : Crosstalk attribute parameter.

17.3.1 ISP_CROSSTALK_MANUAL_ATTR_S

【Description】

Crosstalk attribute manual parameter.

【Syntax】

```
typedef struct _ISP_CROSSTALK_MANUAL_ATTR_S {
   CVI_U16 Strength;
} ISP_CROSSTALK_MANUAL_ATTR_S;
```

【Member】

Member	Description
Strength	G-channel balanced global strength; Value range: [0x0, 0x100] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCrosstalkAttr*
- *CVI_ISP_GetCrosstalkAttr*

17.3.2 ISP_CROSSTALK_AUTO_ATTR_S

【Description】

Crosstalk attribute auto parameter.

【Syntax】

```
typedef struct _ISP_CROSSTALK_AUTO_ATTR_S {
    CVI_U16 Strength[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CROSSTALK_AUTO_ATTR_S;
```

【Member】

Member	Description
Strength	G-channel balanced global strength; Value range: [0x0, 0x100] Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCrosstalkAttr*
- *CVI_ISP_GetCrosstalkAttr*

17.3.3 ISP_CROSSTALK_ATTR_S

【Description】

Crosstalk attribute parameter.

【Syntax】

```
typedef struct _ISP_CROSSTALK_ATTR_S {
    CVI_BOOL Enable; /*RW; Range:[0, 1]*/
    ISP_OP_TYPE_E enOpType;
    CVI_U8 UpdateInterval;
    CVI_U16 GrGbDiffThreSec[4]; /*RW; Range:[0x0, 0xfff]*/
    CVI_U16 FlatThre[4]; /*RW; Range:[0x0, 0xfff]*/
    ISP_CROSSTALK_MANUAL_ATTR_S stManual;
    ISP_CROSSTALK_AUTO_ATTR_S stAuto;
} ISP_CROSSTALK_ATTR_S;
```

【Member】

Member	Description
Enable	The GE module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
GrGbDiffThreSec[4]	G channel balance node threshold 1-4 Value range: [0x0, 0xfff] Data type: CVI_U16
FlatThre[4]	Flat area detection node threshold 1-4 Value range: [0x0, 0xfff] Data type: CVI_U16
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCrosstalkAttr*
- *CVI_ISP_GetCrosstalkAttr*

18 Demosaic

18.1 Function Overview

The image is interpolated from Bayer pattern to complete RGB, and the detail clarity is improved.

18.2 API Reference

- *CVI_ISP_SetDemosaicAttr* : Set demosaic property parameter.
- *CVI_ISP_GetDemosaicAttr* : Get demosaic property parameter.
- *CVI_ISP_SetDemosaicDemoireAttr* : Set demosaic attribute parameter.
- *CVI_ISP_GetDemosaicDemoireAttr* : Get demosaic attribute parameter.

18.2.1 CVI_ISP_SetDemosaicAttr

【Description】

Set demosaic property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetDemosaicAttr(VI_PIPE ViPipe, const ISP_DEMOSAIC_ATTR_S␣
↪*pstDemosaicAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDemosaicAttr	Demosaic attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None..

【Example】

None..

【Related Topic】

- *CVI_ISP_GetDemosaicAttr*

18.2.2 CVI_ISP_GetDemosaicAttr

【Description】

Get Demosaic property parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetDemosaicAttr(VI_PIPE ViPipe, ISP_DEMOSAIC_ATTR_S␣
↪*pstDemosaicAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDemosaicAttr	*Demosaic property parameter*	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDemosaicAttr*

18.2.3 CVI_ISP_SetDemosaicDemoireAttr

【Description】

Set demosaic property parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetDemosaicDemoireAttr(VI_PIPE ViPipe, const ISP_DEMOSAIC_
↪DEMOIRE_ATTR_S *pstDemosaicDemoireAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDemosaicDemoireAttr	Demosaic property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDemosaicDemoireAttr*

18.2.4 CVI_ISP_GetDemosaicDemoireAttr

【Description】

Get Demosaic property parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetDemosaicDemoireAttr(VI_PIPE ViPipe, ISP_DEMOSAIC_DEMOIRE_
↪ATTR_S *pstDemosaicDemoireAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDemosaicDemoireAttr	Demosaic property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDemosaicDemoireAttr*

18.3 Data Types

- *ISP_DEMOSAIC_MANUAL_ATTR_S* : Demosaic attribute manual parameter
- *ISP_DEMOSAIC_AUTO_ATTR_S* : Demosaic attribute automatic parameter
- *ISP_DEMOSAIC_ATTR_S* : Demosaic attribute parameter
- *ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S* : Demosaic Attribute Manual parameter

- *ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S* : Demosaic De-Moire Attribute Auto parameter
- *ISP_DEMOSAIC_DEMOIRE_ATTR_S* : Demosaic Demoir Attribute parameter
- *ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S* : Demosaic Filter Properties Manual parameter
- *ISP_DEMOSAIC_FILTER_AUTO_ATTR_S* : Demosaic Filter Attribute Auto parameter
- *ISP_DEMOSAIC_FILTER_ATTR_S* : Demosaic Filter Attribute parameter

18.3.1 ISP_DEMOSAIC_MANUAL_ATTR_S

【Description】

Demosaic attribute manual parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_MANUAL_ATTR_S {
  CVI_U16 CoarseEdgeThr;
  CVI_U16 CoarseStr;
  CVI_U16 FineEdgeThr;
  CVI_U16 FineStr;
  CVI_U16 RbSigLumaThd;
  CVI_U8 FilterMode;
} ISP_DEMOSAIC_MANUAL_ATTR_S;
```

【Member】

Member	Description
CoarseEdgeThr	Edge coarse tuning detection threshold. The smaller the value is, the more edges are detected. It is recommended to use the parameter CoarseStr for debugging. Value range: [0x0, 0xfff] Data type: CVI_U16
CoarseStr	Rough adjust the intensity value of the edge. The smaller the value is, the more directional the processing is. On the contrary, the more non directional processing Value range: [0x0, 0xfff] Data type: CVI_U16
FineEdgeThr	Edge fine tuning detection threshold. The smaller the value is, the more edges are detected. It is recommended to use the parameter FineStr for debugging Value range: [0x0, 0xfff] Data type: CVI_U16
FineStr	Fine tune the intensity value at the edge. The smaller the value is, the more directional the processing is. On the contrary, the more non directional processing. Value range: [0x0, 0xfff] Data type: CVI_U16
FilterMode	The sharpening width of the DC image, the larger value, the wider width of the sharpened edge, and the more eye-catching it is visually. Value range: [0x0, 0x1] Data Type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicAttr*
- *CVI_ISP_GetDemosaicAttr*

18.3.2 ISP_DEMOSAIC_AUTO_ATTR_S

【Description】

Demosaic attribute automatic parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_AUTO_ATTR_S {
 CVI_U16 CoarseEdgeThr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 CoarseStr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 FineEdgeThr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 FineStr[ISP_AUTO_ISO_STRENGTH_NUM];
```

(continues on next page)

(continued from previous page)

```
  CVI_U16 RbSigLumaThd[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 FilterMode[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DEMOSAIC_AUTO_ATTR_S;
```

【Member】

Member	Description
CoarseEdgeThr	Edge roughness detection threshold.The smaller the value, the more edges are detected.Recommended with parameter CoarseStr debugging 。 Value range: [0x0, 0xfff] Data type: CVI_U16
CoarseStr	Edge roughness intensity value.The smaller the value, the more directional the processing.Conversely, the more disoriented the processing Value range: [0x0, 0xfff] Data type: CVI_U16
FineEdgeThr	Edge fine tuning detection threshold.The smaller the value, the more edges are detected.It is recommended to debug with the parameter FineStr. Value range: [0x0, 0xfff] Data type: CVI_U16
FineStr	Edge fine tune intensity value.The smaller the value, the more directional the processing.Conversely, the more disoriented the processing. Value range: [0x0, 0xfff] Data type: CVI_U16
FilterMode	The sharpening width of the DC image, the larger the value, the wider the width of the sharpened edge, and the more eye-catching it is visually. Value range: [0x0, 0x1] Data Type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicAttr*
- *CVI_ISP_GetDemosaicAttr*

18.3.3 ISP_DEMOSAIC_ATTR_S

【Description】

Demosaic attribute parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_ATTR_S {
 CVI_BOOL Enable;
 CVI_BOOL TuningMode;
 CVI_BOOL RbVtEnable;
 ISP_OP_TYPE_E enOpType;
 CVI_U8 UpdateInterval;
 ISP_DEMOSAIC_MANUAL_ATTR_S stManual;
 ISP_DEMOSAIC_AUTO_ATTR_S stAuto;
} ISP_DEMOSAIC_ATTR_S;
```

【Member】

Member	Description
Enable	Demosaic module enabled. 0:Off. 1:En 《 abled. Value range: [0, 1] Data type: CVI_BOOL
TuningMode	Output debug strategy, auxiliary adjustment parameter. 0: Demosaic image result. 1:Flat/Vertical/Horizontal Edge Detection Image Result. Value range: [0, 1] Data type: CVI_BOOL
enOpType	operation type OP_TYPE_AUTO: Automatic mode OP_TYPE_MANUAL: Manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data Type: CVI_U8
stManual	Manual Mode parameter Properties
stAuto	Automatic Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicAttr*
- *CVI_ISP_GetDemosaicAttr*

18.3.4 ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S

【Description】

Demosaic Demoire Attribute Manual parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S {
   CVI_U8 AntiFalseColorStr; /*RW; Range:[0x0, 0xff]*/
   CVI_U16 SatGainIn[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 SatGainOut[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 ProtectColorGainIn[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 ProtectColorGainOut[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 UserDefineProtectColor1; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 UserDefineProtectColor2; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 UserDefineProtectColor3; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 EdgeGainIn[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 EdgeGainOut[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 DetailGainIn[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 DetailGaintOut[2]; /*RW; Range:[0x0, 0xfff]*/
   CVI_U16 DetailDetectLumaStr; /*RW; Range:[0x0, 0xfff]*/
   CVI_U8 DetailSmoothStr; /*RW; Range:[0x0, 0xff]*/
   CVI_U8 DetailWgtThr; /*RW; Range:[0x0, 0xff]*/
   CVI_U16 DetailWgtMin; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 DetailWgtMax; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 DetailWgtSlope; /*RW; Range:[0x0, 0x400]*/
   CVI_U8 EdgeWgtNp; /*RW; Range:[0x0, 0xff]*/
   CVI_U8 EdgeWgtThr; /*RW; Range:[0x0, 0xff]*/
   CVI_U16 EdgeWgtMin; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 EdgeWgtMax; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 EdgeWgtSlope; /*RW; Range:[0x0, 0x400]*/
   CVI_U8 DetailSmoothMapTh; /*RW; Range:[0x0, 0xff]*/
   CVI_U16 DetailSmoothMapMin; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 DetailSmoothMapMax; /*RW; Range:[0x0, 0x100]*/
   CVI_U16 DetailSmoothMapSlope; /*RW; Range:[0x0, 0x400]*/
   CVI_U8 LumaWgt; /*RW; Range:[0x0, 0xff]*/
} ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S;
```

【Member】

Member	Description
DetailSmoothStr	Strength of detail smoothing.The larger the value, the stronger the smoothing intensity and the stronger the suppression of pseudo-details. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicDemoireAttr*
- *CVI_ISP_GetDemosaicDemoireAttr*

18.3.5 ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S

【Description】

Demosaic De-Moire Attribute Auto parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S {
    CVI_U8 AntiFalseColorStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xff]*/
    CVI_U16 SatGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xfff]*/
    CVI_U16 SatGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xfff]*/
    CVI_U16 ProtectColorGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xfff]*/
    CVI_U16 ProtectColorGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,
↪ 0xfff]*/
    CVI_U16 UserDefineProtectColor1[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xfff]*/
    CVI_U16 UserDefineProtectColor2[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xfff]*/
    CVI_U16 UserDefineProtectColor3[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xfff]*/
    CVI_U16 EdgeGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xfff]*/
    CVI_U16 EdgeGainOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xfff]*/
    CVI_U16 DetailGainIn[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xfff]*/
    CVI_U16 DetailGaintOut[2][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xfff]*/
    CVI_U16 DetailDetectLumaStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xfff]*/
    CVI_U8 DetailSmoothStr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 DetailWgtThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailWgtMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 DetailWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0x400]*/
    CVI_U8 EdgeWgtNp[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 EdgeWgtThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 EdgeWgtMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x100]*/
    CVI_U16 EdgeWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x400]*/
    CVI_U8 DetailSmoothMapTh[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0xff]*/
```

(continues on next page)

(continued from previous page)

```
    CVI_U16 DetailSmoothMapMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0x100]*/
    CVI_U16 DetailSmoothMapMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0x100]*/
    CVI_U16 DetailSmoothMapSlope[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,␣
↪0x400]*/
    CVI_U8 LumaWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
} ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S;
```

【Member】

Member	Description
DetailSmoothStr	Strength of detail smoothing.The larger the value, the stronger the smoothing intensity and the stronger the suppression of pseudo-details. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicDemoireAttr*
- *CVI_ISP_GetDemosaicDemoireAttr*

18.3.6 ISP_DEMOSAIC_DEMOIRE_ATTR_S

【Description】

Demosaic Demoir Attribute parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_DEMOIRE_ATTR_S {
    CVI_BOOL AntiFalseColorEnable; /*RW; Range:[0, 1]*/
    CVI_BOOL ProtectColorEnable; /*RW; Range:[0, 1]*/
    CVI_BOOL DetailDetectLumaEnable; /*RW; Range:[0, 1]*/
    CVI_BOOL DetailSmoothEnable; /*RW; Range:[0, 1]*/
    CVI_BOOL DetailMode; /*RW; Range:[0, 1]*/
    ISP_DEMOSAIC_DEMOIRE_MANUAL_ATTR_S stManual;
    ISP_DEMOSAIC_DEMOIRE_AUTO_ATTR_S stAuto;
} ISP_DEMOSAIC_DEMOIRE_ATTR_S;
```

【Member】

Member	Description
stManual	Manual Mode parameter Properties
stAuto	Automatic Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDemosaicDemoireAttr*
- *CVI_ISP_GetDemosaicDemoireAttr*

18.3.7 ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S

【Description】

Demosaic Filter Properties Manual parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S {
 CVI_U16 DetailWgtThr;
 CVI_U8 DetailWgtSlope;
 CVI_U8 DetailWgtMin;
 CVI_U8 DetailWgtMax;
 CVI_U16 EdgeWgtThr;
 CVI_U8 EdgeWgtSlope;
 CVI_U8 EdgeWgtMin;
 CVI_U8 EdgeWgtMax;
 CVI_U16 LumaWgtThr;
 CVI_U8 LumaWgtSlope;
 CVI_U8 LumaWgtMin;
 CVI_U8 LumaWgtMax;
} ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S;
```

【Member】

Member	Description
DetailWgtThr	Detail retention range threshold.The smaller the value, the larger the range of detail preservation. Value range: [0x0, 0xfff] Data type: CVI_U16
DetailWgtSlope	Detail retention strength.The larger the value, the more detail is preserved. Value range: [0x0, 0xff] Data type: CVI_U8
DetailWgtMin	Edge detail smoothing retains the minimum allowable gain. Value range: [0x0, 0xff] Data type: CVI_U8
DetailWgtMax	Edge detail smoothing retains the maximum allowable gain. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtThr	Brightness detail smoothing range threshold.The smaller the value, the larger the range of detail smoothing. Value range: [0x0, 0xfff] Data type: CVI_U16
LumaWgtSlope	Brightness details smooth edges retain strength.The larger the value, the stronger the detail smoothing intensity. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMin	Minimum gain allowed by brightness detail smoothing intensity. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMax	Maximum gain allowed by brightness detail smoothing intensity. Value range: [0x0, 0xff] Data type: CVI_U8
EdgeWgtThr	Detail preservation range threshold. The smaller the value, the greater the range of detail preservation. Value range: [0x0, 0xfff] Data Type: CVI_U16
EdgeWgtSlope	Details retain intensity. The larger the value, the more detail is preserved. Value range: [0x0, 0xff] Data Type: CVI_U8
EdgeWgtMin	Edge detail smoothing preserves the minimum gain allowed. Value range: [0x0, 0xff] Data Type: CVI_U8
EdgeWgtMax	Maximum gain allowed for edge detail smoothing preservation. Value range: [0x0, 0xff] Data Type: CVI_U8

【Note】

This function of cv180x & cv181x is not supported.

【Related Data Type and Interface】

None.

18.3.8 ISP_DEMOSAIC_FILTER_AUTO_ATTR_S

【Description】

Demosaic Filter Attribute Auto parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_FILTER_AUTO_ATTR_S {
 CVI_U16 DetailWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DetailWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DetailWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DetailWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 EdgeWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 EdgeWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 EdgeWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 EdgeWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 LumaWgtThr[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 LumaWgtSlope[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 LumaWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 LumaWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DEMOSAIC_FILTER_AUTO_ATTR_S;
```

【Member】

Member	Description
DetailWgtThr	Detail retention range threshold.The smaller the value, the larger the range of detail preservation 。 Value range: [0x0, 0xfff] Data type: CVI_U16
DetailWgtSlope	Detail retention strength.The larger the value, the more detail is preserved. Value range: [0x0, 0xff] Data type: CVI_U8
DetailWgtMin	Edge detail smoothing retains the minimum allowable gain. Value range: [0x0, 0xff] Data type: CVI_U8
DetailWgtMax	Edge detail smoothing retains the maximum allowable gain. Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtThr	Brightness detail smoothing range threshold.The smaller the value, the larger the range of detail smoothing. Value range: [0x0, 0xfff] Data type: CVI_U16
LumaWgtSlope	Brightness details smooth edges retain strength.The larger the value, the stronger the detail smoothing intensity Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMin	Minimum gain allowed by brightness detail smoothing intensity Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMax	Maximum gain allowed by brightness detail smoothing intensity Value range: [0x0, 0xff] Data type: CVI_U8
EdgeWgtThr	Detail preservation range threshold. The smaller the value, the greater the range of detail preservation. Value range: [0x0, 0xfff] Data Type: CVI_U16
EdgeWgtSlope	Details retain intensity. The larger the value, the more detail is preserved. Value range: [0x0, 0xff] Data Type: CVI_U8
EdgeWgtMin	Edge detail smoothing preserves the minimum gain allowed. Value range: [0x0, 0xff] Data Type: CVI_U8
EdgeWgtMax	Maximum gain allowed for edge detail smoothing preservation. Value range: [0x0, 0xff] Data Type: CVI_U8

【Note】

This function of cv180x & cv181x is not supported.

【Related Data Type and Interface】

None.

18.3.9 ISP_DEMOSAIC_FILTER_ATTR_S

【Description】

Demosaic Filter Attribute parameter

【Syntax】

```
typedef struct _ISP_DEMOSAIC_FILTER_ATTR_S {
 ISP_OP_TYPE_E enOpType;
 ISP_DEMOSAIC_FILTER_MANUAL_ATTR_S stManual;
 ISP_DEMOSAIC_FILTER_AUTO_ATTR_S stAuto;
} ISP_DEMOSAIC_FILTER_ATTR_S;
```

【Member】

Member	Description
enOpType	operation type OP_TYPE_AUTO: Automatic mode OP_TYPE_MANUAL: manual mode
stManual	Manual Mode parameter Properties
stAuto	Automatic Mode parameter Properties

【Note】

This function of cv180x & cv181x is not supported.

【Related Data Type and Interface】

None.

19 Sharpen

19.1 Function Overview

This module is used to enhance image clarity, after 3DNR, it mainly sharpens the large edges in the image.

19.2 API reference

- *CVI_ISP_SetSharpenAttr* : Set sharpening property parameters
- *CVI_ISP_GetSharpenAttr* : Get sharpening property parameters

19.2.1 CVI_ISP_SetSharpenAttr

【Description】

Set sharpening property parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetSharpenAttr(VI_PIPE ViPipe, const ISP_SHARPEN_ATTR_S␣
↪*pstDRCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstSharpenAttr	Sharpening attribute parameters	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetSharpenAttr*

19.2.2 CVI_ISP_GetSharpenAttr

【Description】

Get sharpening property parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetSharpenAttr(VI_PIPE ViPipe, ISP_SHARPEN_ATTR_S *pstDRCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstSharpenAttr	Sharpening attribute parameters	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetSharpenAttr*

19.3 Data Type

- *ISP_SHARPEN_MANUAL_ATTR_S* : Demosaic Sharpening Properties Manual parameters
- *ISP_SHARPEN_AUTO_ATTR_S* : Demosaic sharpening attribute automatic parameters
- *ISP_SHARPEN_ATTR_S* : Demosaic Sharpening Attribute parameters

19.3.1 ISP_SHARPEN_MANUAL_ATTR_S

【Description】

Demosaic Sharpening Properties Manual parameters

【Syntax】

```
typedef struct _ISP_SHARPEN_MANUAL_ATTR_S {
 CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM];
 CVI_U8 LumaCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 LumaCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 MotionCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
 CVI_U8 GlobalGain; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 HFBlendWgt; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MFBlendWgt; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThr; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThr; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThrMax; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThrMin; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
 CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 SatShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 SatShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
} ISP_SHARPEN_MANUAL_ATTR_S;
```

【Member】

Member	Description
LumaAdpGain[33]	Brightness Sharpening Weight Value range: [0x0, 0x3f] Data type: CVI_U8
LumaCor-LutIn[EE_LUT_NODE]	Based on luma coring, this is the input node, input luma Value range: [0x0, 0xff] Data type: CVI_U8
LumaCor-LutOut[EE_LUT_NODE]	Based on luma' s coring, this is the output node, and the output corresponds to the coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutIn[EE_LUT_NODE]	Motion-based coring, this is the input node, the input is motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutOut[EE_LUT_NODE]	Motion-based coring, this is the output node, output the corresponding coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutIn[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the input node, input the size of motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutOut[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the output node, output the weight of motion coring. (max = 128) Value range: [0x0, 0x80] Data type: CVI_U8
GlobalGain	Global sharpening weight Value range: [0x0, 0xff] Data type: CVI_U8
OverShootGain	White edge sharpening weight Value range: [0x0, 0x3f] Data type: CVI_U8
UnderShootGain	Black edge sharpening weight Value range: [0x0, 0x3f] Data type: CVI_U8
EdgeGlobalGain	Global edge sharpening weights Value range: [0x0, 0x3f] Data type: CVI_U8
DeltaShtCtrl[33]	Define the degree of edge enhancement according to the result of edge detection Value range: [0x0, 0x3f] Data type: CVI_U8
DeltaShtCtrlUnit	Intensity control for edge detection Value range: [0x0, 0x3] Data type: CVI_U8
YNoiseLevel	Luminance sharpening noise value magnification, one time is 64. The larger the value, the more brightness sharpening noise is amplified, and vice versa Value range: [0x0, 0xff] Data type: CVI_U8
HFBlendWgt	Weights for high-frequency edge enhancement Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSharpenAttr*
- *CVI_ISP_GetSharpenAttr*

19.3.2 ISP_SHARPEN_AUTO_ATTR_S

【Description】

Demosaic sharpening attribute automatic parameters

【Syntax】

```
typedef struct _ISP_SHARPEN_MANUAL_ATTR_S {
 CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x3f]*/
 CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 LumaCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,
↪ 0xff]*/
 CVI_U8 LumaCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x20]*/
 CVI_U8 MotionCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x20]*/
 CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x80]*/
 CVI_U8 GlobalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 UnderShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 HFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThrMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThrMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x80]*/
 CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x3f]*/
 CVI_U8 SatShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,
↪ 0xff]*/
 CVI_U8 SatShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x80]*/
```

(continues on next page)

(continued from previous page)

```
  CVI_U8 SatShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;
↪Range:[0x0, 0x80]*/
} ISP_SHARPEN_AUTO_ATTR_S;
```

【Member】

Member	Description
LumaAdpGain[33]	Brightness Sharpening Weight Value range: [0x0, 0x3f] Data type: CVI_U8
LumaCor-LutIn[EE_LUT_NODE]	Based on luma coring, this is the input node, input luma Value range: [0x0, 0xff] Data type: CVI_U8
LumaCor-LutOut[EE_LUT_NODE]	Based on luma' s coring, this is the output node, and the output corresponds to the coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutIn[EE_LUT_NODE]	Motion-based coring, this is the input node, the input is motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutOut[EE_LUT_NODE]	Motion-based coring, this is the output node, output the corresponding coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutIn[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the input node, input the size of motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutOut[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the output node, output the weight of motion coring. (max = 128) Value range: [0x0, 0x80] Data type: CVI_U8
GlobalGain	Global sharpening weight Value range: [0x0, 0xff] Data type: CVI_U8
OverShootGain	White edge sharpening weight Value range: [0x0, 0x3f] Data type: CVI_U8
UnderShootGain	Black edge sharpening weight Value range: [0x0, 0x3f] Data type: CVI_U8
EdgeGlobalGain	Global edge sharpening weights Value range: [0x0, 0x3f] Data type: CVI_U8
DeltaShtCtrl[33]	Define the degree of edge enhancement according to the result of edge detection Value range: [0x0, 0x3f] Data type: CVI_U8
DeltaShtCtrlUnit	Intensity control for edge detection Value range: [0x0, 0x3] Data type: CVI_U8
YNoiseLevel	Luminance sharpening noise value magnification, one time is 64. The larger the value, the more brightness sharpening noise is amplified, and vice versa Value range: [0x0, 0xff] Data type: CVI_U8
HFBlendWgt	Weights for high-frequency edge enhancement Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSharpenAttr*
- *CVI_ISP_GetSharpenAttr*

19.3.3 ISP_SHARPEN_ATTR_S

【Description】

Demosaic Sharpening Attribute parameters

【Syntax】

```
typedef struct _ISP_SHARPEN_ATTR_S {
  CVI_BOOL Enable; /*RW; Range:[0x0, 0x1]*/
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
  CVI_U8 TuningMode; /*RW; Range:[0x0, 0xb]*/
  CVI_BOOL LumaAdpGainEn; /*RW; Range:[0x0, 0x1]*/
  CVI_BOOL DeltaAdpGainEn; /*RW; Range:[0x0, 0x1]*/
  CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
  CVI_BOOL NoiseSuppressEnable; /*RW; Range:[0, 1]*/
  CVI_BOOL SatShtCtrlEn; /*RW; Range:[0, 1]*/
  CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/
  CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/
  CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/
  ISP_SHARPEN_MANUAL_ATTR_S stManual;
  ISP_SHARPEN_AUTO_ATTR_S stAuto;
} ISP_SHARPEN_ATTR_S;
```

【Member】

Member	Description
Enable	Y Sharpen module enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
TuningMode	Output debug strategy, auxiliary adjustment parameters Value range: [0x0, 0xb] Data type: CVI_U8
LumaAdpCoringEn	Automatic Luminance Noise Rejection Threshold Switch Value range: [0, 1] Data type: CVI_BOOL
LumaAdpGainEn	Brightness sharpening weights enabled Value range: [0, 1] Data type: CVI_BOOL
DeltaAdpGainEn	sharpen weights enable Value range: [0, 1] Data type: CVI_BOOL
DeltaAdpGain[SHARPEN_LUT_NUM]	sharpening weight Value range: [0x0, 0x3f] Data type: CVI_U8
NoiseSuppressEnable	Enhanced denoising enable. For edge detection images, do pre-processing to strengthen denoising and then do edge enhancement Value range: [0, 1] Data type: CVI_BOOL
SatShtCtrlEn	Enablement of edge enhancement adjusted by saturation enable Value range: [0, 1] Data type: CVI_BOOL
SoftClampEnable	Smooth Processing Edge Enhancement Value range: [0, 1] Data type: CVI_BOOL
SoftClampUB	The upper bound of edge enhancement for smooth processing. The larger the value is, the more continuous the edge enhancement will be, but the weaker the enhancement effect will be. Value range: [0, 0xff] Data type: CVI_U8
SoftClampLB	The lower bound of edge enhancement for smooth processing. The larger the value is, the more continuous the edge enhancement will be, but the weaker the enhancement effect will be. Value range: [0, 0xff] Data type: CVI_U8
stManual	Manual Mode parameter Properties
stAuto	Auto Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSharpenAttr*
- *CVI_ISP_GetSharpenAttr*

20 PreSharpen

20.1 Function Overview

This module is used to enhance image sharpness, pre-3DNR, and focuses on sharpening directional edges and detailed textures in images.

20.2 API Reference

- *CVI_ISP_SetPreSharpenAttr* : Sets the sharpening attribute parameter
- *CVI_ISP_GetPreSharpenAttr* : Obtains the sharpening attribute parameter

20.2.1 CVI_ISP_SetPreSharpenAttr

【Description】

Sets the sharpening attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetPreSharpenAttr(VI_PIPE ViPipe, const ISP_PRESHARPEN_ATTR_S *pstPreSharpenAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstPreSharpenAttr	Sharpening attribute parameters	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetPreSharpenAttr*

20.2.2 CVI_ISP_GetPreSharpenAttr

【Description】

Obtains the sharpening attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetPreSharpenAttr(VI_PIPE ViPipe, ISP_PRESHARPEN_ATTR_S
*pstPreSharpenAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstPreSharpenAttr	Sharpening attribute parameters	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetPreSharpenAttr*

20.3 Data Types

- *ISP_PRESHARPEN_MANUAL_ATTR_S* : sharpening attribute manual parameter
- *ISP_PRESHARPEN_AUTO_ATTR_S* : sharpening attribute automatic parameter
- *ISP_PRESHARPEN_ATTR_S* : sharpening attribute parameter

20.3.1 ISP_PRESHARPEN_MANUAL_ATTR_S

【Description】

sharpening attribute manual parameter

【Syntax】

```
typedef struct _ISP_PRESHARPEN_MANUAL_ATTR_S {
 CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 LumaCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 LumaCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 MotionCorLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionCorLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
 CVI_U8 GlobalGain; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 HFBlendWgt; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MFBlendWgt; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThr; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThr; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThrMax; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThrMin; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
 CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 SatShtGainIn[EE_LUT_NODE]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 SatShtGainOut[EE_LUT_NODE]; /*RW; Range:[0x0, 0x80]*/
} ISP_PRESHARPEN_MANUAL_ATTR_S;
```

【Member】

Member	Description
EdgeGain	Edge Enhancement parameters The larger the value, the greater the edge sharpening strength Value range: [0x0, 0x3f] Data type: CVI_U8
TextureGain	Enhancement parameters for detail textures The larger the value, the greater the detail texture sharpening intensity Value range: [0x0, 0x3ff] Data type: CVI_U16
EdgeFreq	Image Directional Edge Band Control Value range: [0x0, 0x4] Data type: CVI_U8
TextureFreq	Image non-directional detail texture band control Value range: [0x0, 0x4] Data type: CVI_U8
LumaAdpCoring[33]	Luminance sharpening noise values, detail texture or edge enhancement will exclude the enhancement contributed by this tolerance value. The smaller the value, the more obvious the effect of sharpening, but the easier it is to sharpen the noise. The larger the value is, the less likely it is to sharpen the noise, but the sharpening effect is less obvious Value range: [0x0, 0x3ff] Data type: CVI_U16
LumaAdpGain[33]	Brightness Sharpening Weight Value range: [0x0, 0x3f] Data type: CVI_U8
LumaCor-LutIn[EE_LUT_NODE]	Based on luma coring, this is the input node, input luma Value range: [0x0, 0xff] Data type: CVI_U8
LumaCor-LutOut[EE_LUT_NODE]	Based on luma’ s coring, this is the output node, and the output corresponds to the coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutIn[EE_LUT_NODE]	Motion-based coring, this is the input node, the input is motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCor-LutOut[EE_LUT_NODE]	Motion-based coring, this is the output node, output the corresponding coring value Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutIn[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the input node, input the size of motion Value range: [0x0, 0xff] Data type: CVI_U8
MotionCorWgt-LutOut[EE_LUT_NODE]	Adjust the weight of luma coring and motion coring based on motion, this is the output node, output the weight of motion coring. (max = 128) Value range: [0x0, 0xff] Data type: CVI_U8
GlobalGain	Global sharpening weight Value range: [0x0, 0xff]
	Data type: CVI_U8
EdgeGlobalGain	Global edge sharpening weights Value range: [0x0, 0x3f]

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetPreSharpenAttr*
- *CVI_ISP_GetPreSharpenAttr*

20.3.2 ISP_PRESHARPEN_AUTO_ATTR_S

【Description】

sharpening attribute automatic parameter

【Syntax】

```
typedef struct _ISP_PRESHARPEN_AUTO_ATTR_S {
 CVI_U8 LumaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x3f]*/
 CVI_U8 DeltaAdpGain[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x3f]*/
 CVI_U8 LumaCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,
↪ 0xff]*/
 CVI_U8 LumaCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x20]*/
 CVI_U8 MotionCorLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionCorLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x20]*/
 CVI_U8 MotionCorWgtLutIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionCorWgtLutOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x80]*/
 CVI_U8 GlobalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 UnderShootGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0x3f]*/
 CVI_U8 HFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MFBlendWgt[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThr[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 OverShootThrMax[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 UnderShootThrMin[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0xff]*/
 CVI_U8 MotionShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x80]*/
 CVI_U8 HueShtCtrl[SHARPEN_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;␣
↪Range:[0x0, 0x3f]*/
 CVI_U8 SatShtGainIn[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0,
↪ 0xff]*/
```

(continues on next page)

(continued from previous page)

```
  CVI_U8 SatShtGainOut[EE_LUT_NODE][ISP_AUTO_ISO_STRENGTH_NUM]; /*RW;
↪Range:[0x0, 0x80]*/
} ISP_PRESHARPEN_AUTO_ATTR_S;
```

【Member】

Member	Description
EdgeGain	Edge Enhancement parameters The larger the value, the greater the edge sharpening strength Value range: [0x0, 0x3f] Data type: CVI_U8
OverShootThr	White edge sharpening upper limit Value range: [0x0, 0xff] Data type: CVI_U8
UnderShootThr	Lower limit of black edge sharpening Value range: [0x0, 0xff] Data type: CVI_U8
YNoiseLevel	Luminance sharpening noise value magnification, one time is 64. The larger the value, the more brightness sharpening noise is amplified, and vice versa Value range: [0x0, 0xff] Data type: CVI_U8
OverShootThrMax	White edge sharpening maximum upper limit range Value range: [0x0, 0xff] Data type: CVI_U8
UnderShootThrMin	Black edge sharpening maximum and minimum range Value range: [0x0, 0xff] Data type: CVI_U8
MotionShtGainIn[4]	For the LUT that determines the degree of edge enhancement in the motion area, this is a horizontal node, and the input value is the motion value Value range: [0x0, 0xff] Data type: CVI_U8
MotionShtGainOut[4]	For the LUT that determines the degree of edge enhancement in the motion area, this is a vertical node, and the output value is the enhancement strength corresponding to the motion Value range: [0x0, 0xff] Data type: CVI_U8
HueShtCtrl[SHARPEN_LUT_NUM]	Edge enhancement based on specified color Value range: [0x0, 0x3f] Data type: CVI_U8
SatShtGainIn[EE_LUT_NODE]	Edge enhancement based on the specified saturation, this is the input node, input saturation Value range: [0x0, 0xff] Data type: CVI_U8
SatShtGainOut[EE_LUT_NODE]	Edge enhancement based on the specified saturation, this is the output node, which outputs the edge strength corresponding to the saturation. (max = 128) Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetPreSharpenAttr*
- *CVI_ISP_GetPreSharpenAttr*

20.3.3 ISP_PRESHARPEN_ATTR_S

【Description】

sharpening attribute parameter

【Syntax】

```
typedef struct _ISP_PRESHARPEN_ATTR_S {
 CVI_BOOL Enable; /*RW; Range:[0x0, 0x1]*/
 ISP_OP_TYPE_E enOpType;
 CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
 CVI_U8 TuningMode; /*RW; Range:[0x0, 0xb]*/
 CVI_BOOL LumaAdpGainEn; /*RW; Range:[0x0, 0x1]*/
 CVI_BOOL DeltaAdpGainEn; /*RW; Range:[0x0, 0x1]*/
 CVI_BOOL NoiseSuppressEnable; /*RW; Range:[0, 1]*/
 CVI_BOOL SatShtCtrlEn; /*RW; Range:[0, 1]*/
 CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/
 CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/
 ISP_PRESHARPEN_MANUAL_ATTR_S stManual;
 ISP_PRESHARPEN_AUTO_ATTR_S stAuto;
} ISP_PRESHARPEN_ATTR_S;
```

【Member】

Member	Description
Enable	Y Sharpen module enabled Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
TuningMode	Output debug strategy, auxiliary adjustment parameters Value range: [0x0, 0xb] Data type: CVI_U8
TuningModeLevelShift	Auxiliary Tuning parameter Gain for TuningMode 1, 2, and 3 Value range: [0x0, 0x8] Data type: CVI_U8
FilterCoefByISOFuncEn	Dynamic filter coefficient enable Value range: [0, 1] Data type: CVI_BOOL
FilterCoefByISOLowThd	The lower threshold of the dynamic filter coefficient, below this value, it is preferred to use the directional filter coefficient Value range: [0, 0xffffffff] Data type: CVI_U32
FilterCoefByISOHighThd	The upper threshold of the dynamic filter coefficient, below this value it is preferred to use the undirected filter coefficient Value range: [0, 0xffffffff] Data type: CVI_U32
FilterCoefNonDirType	Dynamic filter coefficient category Value range: [0, 0xff] Data type: CVI_U32
LumaAdpCoringEn	Automatic Luminance Noise Rejection Threshold Switch Value range: [0, 1] Data type: CVI_BOOL
LumaAdpGainEn	Brightness sharpening weights enabled Value range: [0, 1] Data type: CVI_BOOL
DeltaAdpGainEn	sharpen weights enabled Value range: [0, 1] Data type: CVI_BOOL
WdrCoringCompensationEn	Luminance sharpening noise value compensation is enabled in wdr mode. Value range: [0, 1] Data type: CVI_BOOL
WdrCoringCompensation-Mode	Compensation mode for luminance sharpening noise value in wdr mode. 0: Compensate luminance sharpening noise according to Wdr-CoringHighThrd and WdrCoringLowthd. 1: Automatically compensate luminance sharpening noise according to DRC tone mapping curve. Value range: [0, 1]
	Data type: CVI_BOOL
WdrCoringToleranceLevel	Tolerance value for luma sharpening noise compensation in wdr mode. The smaller the value, the more obvious the effect

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetPreSharpenAttr*
- *CVI_ISP_GetPreSharpenAttr*

21 Gamma

21.1 Function Overview

Because the human eye is more sensitive to low brightness, it does nonlinear conversion of the brightness of the image, compresses the bright area and stretches the dark area to make the overall brightness feel better.

21.2 API Reference

- *CVI_ISP_SetGammaAttr* : Set the gamma attribute parameter
- *CVI_ISP_GetGammaAttr* : Get the gamma attribute parameter
- *CVI_ISP_GetGammaCurveByType* : Get Gamma curve information
- *CVI_ISP_SetAutoGammaAttr* : Set Auto Gamma attribute parameters
- *CVI_ISP_GetAutoGammaAttr* : Get Auto Gamma attribute parameters

21.2.1 CVI_ISP_SetGammaAttr

【Description】

- Set the gamma attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetGammaAttr(VI_PIPE ViPipe, const ISP_GAMMA_ATTR_S␣
↪*pstGammaAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstGammaAttr	Gamma attribute parameters	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetGammaAttr*

21.2.2 CVI_ISP_GetGammaAttr

【Description】

Get the gamma attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetGammaAttr(VI_PIPE ViPipe, ISP_GAMMA_ATTR_S *pstGammaAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstGammaAttr	Gamma attribute parameters	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetGammaAttr*

21.2.3 CVI_ISP_GetGammaCurveByType

【Description】

- Get Gamma curve information

【Syntax】

```
CVI_S32 CVI_ISP_GetGammaCurveByType(VI_PIPE ViPipe, ISP_GAMMA_ATTR_S␣
↪*pstGammaAttr, const ISP_GAMMA_CURVE_TYPE_E curveType);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstGammaAttr	Gamma property parameter	Output
curveType	Gamma Curve Type	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetGammaAttr*
- *CVI_ISP_GetGammaAttr*

21.2.4 CVI_ISP_SetAutoGammaAttr

【Description】

Set Auto Gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetAutoGammaAttr(VI_PIPE ViPipe, const ISP_AUTO_GAMMA_ATTR_S␣
↪*pstGammaAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstGammaAttr	Auoto Gamma attribute parameters	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetAutoGammaAttr*

21.2.5 CVI_ISP_GetAutoGammaAttr

【Description】

- Get Auto Gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetAutoGammaAttr(VI_PIPE ViPipe, ISP_AUTO_GAMMA_ATTR_S␣
↪*pstGammaAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstGammaAttr	Auoto Gamma attribute parameters	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetAutoGammaAttr*

21.3 Data Types

- *ISP_GAMMA_ATTR_S* : Gamma attribute parameter
- *ISP_GAMMA_CURVE_ATTR_S* : Auto Gamma curve parameters
- *ISP_AUTO_GAMMA_ATTR_S* : Auto Gamma attribute parameter

21.3.1 ISP_GAMMA_CURVE_TYPE_E

【Description】

Auto Gamma curve parameters

【Syntax】

```
typedef enum _ISP_GAMMA_CURVE_TYPE_E {
 ISP_GAMMA_CURVE_DEFAULT,
 ISP_GAMMA_CURVE_SRGB,
 ISP_GAMMA_CURVE_USER_DEFINE,
```

(continues on next page)

(continued from previous page)

```
  ISP_GAMMA_CURVE_AUTO,
  ISP_GAMMA_CURVE_MAX
} ISP_GAMMA_CURVE_TYPE_E;
```

【Member】

Member	Description
ISP_GAMMA_CURVE_DEFAULT	default curve
ISP_GAMMA_CURVE_SRGB	SRGB curve
ISP_GAMMA_CURVE_USER_DEFINE	user defined curve
ISP_GAMMA_CURVE_AUTO	Adaptive curve please refer to CVI_ISP_SetAutoGammaAttr CVI_ISP_GetAutoGammaAttr
ISP_GAMMA_CURVE_MAX	

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetGammaAttr*
- *CVI_ISP_GetGammaAttr*
- *CVI_ISP_GetGammaCurveByType*

21.3.2 ISP_GAMMA_ATTR_S

【Description】

Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_GAMMA_ATTR_S {
  CVI_BOOL Enable;
  CVI_U8 UpdateInterval;
  CVI_U16 Table[GAMMA_NODE_NUM];
  ISP_GAMMA_CURVE_TYPE_E enCurveType;
} ISP_GAMMA_ATTR_S;
```

【Member】

Member	Description
Enable	The Gamma function is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
Table[GAMMA_NODE_NUM]	Gamma curve node value Value range: [0x0, 0xfff] Data type: CVI_U16
enCurveType	Gamma Curve Type

【Note】

GAMMA_NODE_NUM = 256

【Related Data Type and Interface】

- *CVI_ISP_SetGammaAttr*
- *CVI_ISP_GetGammaAttr*
- *CVI_ISP_GetGammaCurveByType*

21.3.3 ISP_GAMMA_CURVE_ATTR_S

【Description】

Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_GAMMA_CURVE_ATTR_S {
  CVI_S16 Lv;
  CVI_U16 Tbl[GAMMA_NODE_NUM];
} ISP_GAMMA_CURVE_ATTR_S;
```

【Member】

Member	Description
Lv	Gamma curve corresponds to Lv range Value range: [-500, 1500] Data type: CVI_S16
Tbl[GAMMA_NODE_NUM]	Gamma curve node value Value range: [0x0, 0xfff] Data type: CVI_U16

【Note】

GAMMA_NODE_NUM = 256

【Related Data Type and Interface】

- *CVI_ISP_SetAutoGammaAttr*
- *CVI_ISP_GetAutoGammaAttr*

21.3.4 ISP_AUTO_GAMMA_ATTR_S

【Description】

Auto Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_AUTO_GAMMA_ATTR_S {
  CVI_U8 GammaTabNum;
  ISP_GAMMA_CURVE_ATTR_S
  GammaTab[GAMMA_MAX_INTERPOLATION_NUM];
} ISP_AUTO_GAMMA_ATTR_S;
```

【Member】

Member	Description
GammaTabNum	Number of Adaptive Gamma Curves Value range: [0, 5] Data type: CVI_U8
GammaTab[GAMMA_MAX_INTERPOLATION_NUM]	Adaptive Gamma Curve Information

【Note】

GAMMA_MAX_INTERPOLATION_NUM=5

【Related Data Type and Interface】

- *CVI_ISP_SetAutoGammaAttr*
- *CVI_ISP_GetAutoGammaAttr*

22 DCI

22.1 Function Overview

Dynamic contrast improvement is the abbreviation of dynamic contrast improvement. By adjusting the histogram to enhance the image contrast and enhance the details of the dark part.

22.2 API Reference

- *CVI_ISP_SetDCIAttr* : Set DCI parameter properties
- *CVI_ISP_GetDCIAttr* : Get DCI parameter properties

22.2.1 CVI_ISP_SetDCIAttr

【Description】

Set DCI parameter properties

【Syntax】

```
CVI_S32 CVI_ISP_SetDCIAttr(VI_PIPE ViPipe, const ISP_DCI_ATTR_S *pstDCIAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDCIAttr	*DCI parameter properties*	Input

【Retrun Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDCIAttr*

22.2.2 CVI_ISP_GetDCIAttr

【Description】

Get DCI parameter properties

【Syntax】

```
CVI_S32 CVI_ISP_GetDCIAttr(VI_PIPE ViPipe, ISP_DCI_ATTR_S *pstDCIAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDCIAttr	DCI parameter properties	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDCIAttr*

22.3 Data Types

- *ISP_DCI_MANUAL_ATTR_S* : DCI parameter Manual Properties
- *ISP_DCI_AUTO_ATTR_S* : DCI parameter Auto Properties
- *ISP_DCI_ATTR_S* : DCI parameter Properties

22.3.1 ISP_DCI_MANUAL_ATTR_S

【Description】

DCI parameter Manual Properties

【Syntax】

```
typedef struct _ISP_DCI_MANUAL_ATTR_S {
  CVI_U16 ContrastGain;
  CVI_U8 BlcThr;
  CVI_U8 WhtThr;
  CVI_U16 BlcCtrl;
  CVI_U16 WhtCtrl;
  CVI_U16 DciGainMax;
} ISP_DCI_MANUAL_ATTR_S;
```

【Member】

Member	Description
ContrastGain	It is used to control the intensity of DCI. The higher the value is, the greater the contrast is Value range: [0x0, 0x100] Data type: CVI_U16
BlcThr	The threshold used to determine the range of dark areas. The larger the value is, the larger the range of dark areas is. Value range: [0x0, 0xff] Data type: CVI_U8
WhtThr	The threshold used to determine the range of bright areas. The smaller the value is, the larger the range of bright areas is. Value range: [0x0, 0xff] Data type: CVI_U8
BlcCtrl	Used to determine the contrast of dark areas. When the value is 256, the contrast of dark area remains unchanged. When the ratio is larger than 256, the larger the value is, the greater the contrast in the dark area is; On the contrary, the smaller the value is, the smaller the contrast is. Value range: [0x0, 0x7fff] Data type: CVI_U16
WhtCtrl	Used to determine the contrast of bright areas.When the value is 256, the brightness contrast is unchanged.When it is larger than 256, the larger the value, the greater the contrast of bright area.Conversely, the smaller the value, the smaller the contrast in the bright area. Value range: [0x0, 0x7fff] Data type: CVI_U16
DciGainMax	Used to control the upper limit of contrast. The larger the value, the stronger the contrast. Value range: 0x0, 0x100 Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDCIAttr*
- *CVI_ISP_GetDCIAttr*

22.3.2 ISP_DCI_AUTO_ATTR_S

【Description】

DCI parameter Auto Properties

【Syntax】

```
typedef struct _ISP_DCI_AUTO_ATTR_S {
  CVI_U16 ContrastGain[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BlcThr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 WhtThr[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 BlcCtrl[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 WhtCtrl[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 DciGainMax[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DCI_AUTO_ATTR_S;
```

【Member】

Member	Description
ContrastGain	It is used to control the intensity of DCI. The larger the value is, the greater the contrast is. Value range: [0x0, 0x100] Data type: CVI_U16
BlcThr	The threshold used to determine the range of dark areas. The larger the value, the larger the range of dark areas. Value range: [0x0, 0xff] Data type: CVI_U8
WhtThr	Threshold used to determine the range of the bright area. The smaller the value, the larger the range of the light areas included. Value range: [0x0, 0xff] Data type: CVI_U8
BlcCtrl	Used to determine the contrast of dark areas. If the value is 256, the contrast of dark area remains unchanged. If the value is larger than 256, the larger the value, the greater the contrast in the dark area is; On the contrary, the smaller the value is, the smaller the contrast is. Value range: [0x0, 0x7fff] Data type: CVI_U16
WhtCtrl	Used to determine the contrast of the bright area. When the value is 256, the contrast of the bright area remains unchanged. When the value is larger than 256, the higher the value is, the greater the contrast is; On the contrary, the smaller the value is, the smaller the contrast is. Value range: [0x0, 0x7fff] Data type: CVI_U16
DciGainMax	Used to control the upper limit of contrast. The larger the value, the stronger the contrast. Value range: 0x0, 0x100 Data type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDCIAttr*
- *CVI_ISP_GetDCIAttr*

22.3.3 ISP_DCI_ATTR_S

【Description】

DCI parameter Properties

【Syntax】

```
typedef struct _ISP_DCI_ATTR_S {
  CVI_BOOL Enable;
  CVI_BOOL TuningMode;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_U8 Method;
  CVI_U32 Speed;
  CVI_U16 DciStrength;
  CVI_U16 DciGamma;
  CVI_U8 DciOffset;
  CVI_U8 ToleranceY;
  CVI_U8 Sensitivity;
  ISP_DCI_MANUAL_ATTR_S stManual;
  ISP_DCI_AUTO_ATTR_S stAuto;
} ISP_DCI_ATTR_S;
```

【Member】

Member	Description
Enable	DCI module enable 0: shut down 1: enable
TuningMode	The original image and the result image are displayed at the same time on the same screen to assist in adjusting parameters. 0: CAC image results. 1: The original image and the result image are displayed on the same screen at the same time. Value range: [0, 1] Data type: CVI_BOOL
enOpType	Working type OP_TYPE_AUTO: auto mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
Method	Used to select DCI algo version, 0: adaptive version, 1: manually adjust version Value range: [0x0, 0x1] Data type: CVI_U8
Speed	Smooth intensity, the higher the value, the slower the change. Value range: [0x0, 0x400] Data type: CVI_U16
DciStrength	Keep the lowest slope of each BIN. The larger the value is, the smaller the slope is Value range: [0x0, 0x100] Data type: CVI_U16
DciGamma	It is used to control the contrast. The larger the value, the greater the contrast, but the brightness of the screen will decrease. It is recommended to adjust it together with DciOffset. Value range: [0x64, 0x320] Data type: CVI_U16
DciOffset	Adjustment of overall screen brightness. The larger the value, the greater the overall brightness. It is recommended to adjust together with DciGamma. Value range: [0x0, 0xff] Data type: CVI_U8
ToleranceY	Tolerable luminance error to reach target luminance Value range: [0x0, 0xff] Data type: CVI_U8
Sensitivity	Sensitivity: The higher the sensitivity, the lower the threadhold for dci heavy operations Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Manual mode parameter properties
stAuto	Automatic mode parameter properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDCIAttr*
- *CVI_ISP_GetDCIAttr*

23 LDCI

23.1 Function Overview

Local Dynamic Contrast Improvement is a method to enhance image contrast and improve dark area details by adjusting histogram.

23.2 API Reference

- *CVI_ISP_SetLDCIAttr* : Set LDCI parameter attribute
- *CVI_ISP_GetLDCIAttr* : Get LDCI parameter attribute

23.2.1 CVI_ISP_SetLDCIAttr

【Description】

Set LDCI parameter attribute

【Syntax】

```
CVI_S32 CVI_ISP_SetLDCIAttr(VI_PIPE ViPipe, const ISP_LDCI_ATTR_S *pstLDCIAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstLDCIAttr	DCI parameter properties	Input

【Retrun Value】

Return Value	Description
0	Success
non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetLDCIAttr*

23.2.2 CVI_ISP_GetLDCIAttr

【Description】

Get LDCI parameter attribute

【Syntax】

```
CVI_S32 CVI_ISP_GetLDCIAttr(VI_PIPE ViPipe, ISP_LDCI_ATTR_S *pstLDCIAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstLDCIAttr	DCI parameter properties	Output

【Retrun Value】

Return Value	Description
0	Success
non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetLDCIAttr*

23.3 Data Types

- *ISP_LDCI_GAUSS_COEF_ATTR_S* : LDCI Gaussian structure parameter attribute
- *ISP_LDCI_MANUAL_ATTR_S* : LDCI parameter manual attribute
- *ISP_LDCI_AUTO_ATTR_S* : LDCI parameter automatic attribute
- *ISP_LDCI_ATTR_S* : LDCI parameter attribute

23.3.1 ISP_LDCI_GAUSS_COEF_ATTR_S

【Description】

- LDCI Gaussian structure parameter attribute

【Syntax】

```
typedef struct _ISP_LDCI_GAUSS_COEF_ATTR_S {
  CVI_U8 Wgt;
  CVI_U8 Sigma;
  CVI_U8 Mean;
} ISP_LDCI_GAUSS_COEF_ATTR_S;
```

【Member】

Member	Description
Wgt	Control the LDCI effect intensity according to the brightness, and determine the weight curve of the fusion of the LDCI result and the original image Value range: [0x0, 0x80] Data type: CVI_U8
Sigma	Control the LDCI effect intensity according to the brightness, and determine the weight curve of the fusion of the LDCI result and the original image Value range: [0x0, 0xff] Data type: CVI_U8
Mean	Control the LDCI effect intensity according to the brightness, and determine the weight curve of the fusion of the LDCI result and the original image Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

None

23.3.2 ISP_LDCI_MANUAL_ATTR_S

【Description】

- LDCI parameter manual attribute

【Syntax】

```
typedef struct _ISP_LDCI_MANUAL_ATTR_S {
  CVI_U16 LdciStrength;
  CVI_U16 LdciRange;
  CVI_U16 TprCoef;
  CVI_U8 EdgeCoring;
  CVI_U8 LumaWgtMax;
  CVI_U8 LumaWgtMin;
  CVI_U8 VarMapMax;
  CVI_U8 VarMapMin;
  CVI_U8 UvGainMax;
  CVI_U8 UvGainMin;
  CVI_U8 BrightContrastHigh;
  CVI_U8 BrightContrastLow;
  CVI_U8 DarkContrastHigh;
  CVI_U8 DarkContrastLow;
  ISP_LDCI_GAUSS_COEF_ATTR_S LumaPosWgt;
} ISP_LDCI_MANUAL_ATTR_S;
```

【Member】

Member	Description
LdciStrength	Controls the LDCI enhancement effect parameters. The larger the value, the | stronger the local contrast stretch Value range: [0x0, 0x100] Data type: CVI_U16
LdciRange	Controls the degree of contrast enhancement for high frequency areas of the image. The larger the value, the stronger the contrast of the high frequency area of the image Value range: [0x0, 0x3ff] Data type: CVI_U16
TprCoef	The smoothness of changes in the time domain of the LDCI curve. The smaller the value, the smoother the change in the time domain, and vice versa, the faster the change Value range: [0x0, 0x3ff] Data type: CVI_U16
EdgeCoring	Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMax	Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMin	Value range: [0x0, 0xff] Data type: CVI_U8
VarMapMax	Value range: [0x0, 0xff] Data type: CVI_U8
VarMapMin	Value range: [0x0, 0xff] Data type: CVI_U8
UvGainMax	Value range: [0x0, 0x7f] Data type: CVI_U8
UvGainMin	Value range: [0x0, 0x7f] Data type: CVI_U8
BrightContrastHigh	Value range: [0x0, 0xff] Data type: CVI_U8
BrightContrastLow	Value range: [0x0, 0xff] Data type: CVI_U8
DarkContrastHigh	Value range: [0x0, 0xff] Data type: CVI_U8
DarkContrastLow	Value range: [0x0, 0xff] Data type: CVI_U8
LumaPosWgt	Control the LDCI effect intensity according to the brightness, and determine the weight curve of the fusion of the LDCI result and the original image

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLDCIAttr*
- *CVI_ISP_GetLDCIAttr*

23.3.3 ISP_LDCI_AUTO_ATTR_S

【Description】

LDCI parameter automatic attribute

【Syntax】

```
typedef struct _ISP_LDCI_AUTO_ATTR_S {
  CVI_U16 LdciStrength[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 LdciRange[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 TprCoef[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LumaWgtMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 LumaWgtMin[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 VarMapMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 VarMapMin[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 UvGainMax[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 UvGainMin[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BrightContrastHigh[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 BrightContrastLow[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DarkContrastHigh[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DarkContrastLow[ISP_AUTO_ISO_STRENGTH_NUM];
  ISP_LDCI_GAUSS_COEF_ATTR_S LumaPosWgt[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_LDCI_AUTO_ATTR_S;
```

【Member】

Member	Description
LdciStrength	Controls the LDCI enhancement effect parameters. The larger the value, the	stronger the local contrast stretch Value range: [0x0, 0x100] Data type: CVI_U16
LdciRange	Controls the degree of contrast enhancement for high frequency areas of the image. The larger the value, the stronger the contrast of the high frequency area of the image Value range: [0x0, 0x3ff] Data type: CVI_U16
TprCoef	The smoothness of changes in the time domain of the LDCI curve. The smaller the value, the smoother the change in the time domain, and vice versa, the faster the change Value range: [0x0, 0x3ff] Data type: CVI_U16
EdgeCoring	Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMax	Value range: [0x0, 0xff] Data type: CVI_U8
LumaWgtMin	Value range: [0x0, 0xff] Data type: CVI_U8
VarMapMax	Value range: [0x0, 0xff] Data type: CVI_U8
VarMapMin	Value range: [0x0, 0xff] Data type: CVI_U8
UvGainMax	Value range: [0x0, 0x7f] Data type: CVI_U8
UvGainMin	Value range: [0x0, 0x7f] Data type: CVI_U8
BrightContrastHigh	Value range: [0x0, 0xff] Data type: CVI_U8
BrightContrastLow	Value range: [0x0, 0xff] Data type: CVI_U8
DarkContrastHigh	Value range: [0x0, 0xff] Data type: CVI_U8
DarkContrastLow	Value range: [0x0, 0xff] Data type: CVI_U8
LumaPosWgt	Control the LDCI effect intensity according to the brightness, and determine the weight curve of the fusion of the LDCI result and the original image

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLDCIAttr*
- *CVI_ISP_GetLDCIAttr*

23.3.4 ISP_LDCI_ATTR_S

【Description】

LDCI parameter attribute

【Syntax】

```
typedef struct _ISP_LDCI_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_U8 GaussLPFSigma;
  ISP_LDCI_MANUAL_ATTR_S stManual;
  ISP_LDCI_AUTO_ATTR_S stAuto;
} ISP_LDCI_ATTR_S;
```

【Member】

Member	Description
Enable	LDCI module enable
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
GaussLPFSigma	Local filtering degree, the smaller the value, the more localized the local contrast enhancement effect, and vice versa, the more globalized Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Manual Mode parameter Properties
stAuto	Auto Mode parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLDCIAttr*
- *CVI_ISP_GetLDCIAttr*

24 Dehaze

24.1 Function Overview

Calculate the concentration of fog to automatically adjust the contrast and saturation and achieve the effect of dehazing.

24.2 API Reference

- *CVI_ISP_SetDehazeAttr* : Set dehaze attribute parameters
- *CVI_ISP_GetDehazeAttr* : Get dehaze attribute parameters

24.2.1 CVI_ISP_SetDehazeAttr

【Description】

Set dehaze attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetDehazeAttr(VI_PIPE ViPipe, const ISP_DEHAZE_ATTR_S␣
↪*pstDehazeAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDehazeAttr	Dehaze attribute parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

```
// Enable Dehaze and set strength as 100
VI_PIPE ViPipe = 0;
ISP_DEHAZE_ATTR_S attr;

CVI_ISP_GetDehazeAttr(ViPipe, &attr);
attr.Enable = CVI_TRUE;
attr.enOpType = OP_TYPE_AUTO;
attr. stAuto.Strength[0 /*ISO 100*/] = 100; // Strength=0-100
attr. stAuto.Strength[1 /*ISO 200*/] = 100; // Strength=0-100
...
attr. stAuto.Strength[15 /*ISO 3276800*/] = 100; // Strength=0-100
CVI_ISP_SetDehazeAttr(ViPipe, &attr);
```

【Related topic】

- *CVI_ISP_GetDehazeAttr*

24.2.2 CVI_ISP_GetDehazeAttr

【Description】

Get dehaze attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetDehazeAttr(VI_PIPE ViPipe, ISP_DEHAZE_ATTR_S *pstDehazeAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDehazeAttr	Dehaze attribute parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related topic】

- *CVI_ISP_SetDehazeAttr*

24.3 Data Types

- *ISP_DEHAZE_MANUAL_ATTR_S* : Dehaze attribute manual parameters
- *ISP_DEHAZE_AUTO_ATTR_S* : Dehaze attribute automatic parameters
- *ISP_DEHAZE_ATTR_S* : Dehaze attribute parameters

24.3.1 ISP_DEHAZE_MANUAL_ATTR_S

【Description】

Dehaze attribute manual parameters

【Syntax】

```
typedef struct _ISP_DEHAZE_MANUAL_ATTR_S {
  CVI_U8 Strength;
} ISP_DEHAZE_MANUAL_ATTR_S;
```

【Member】

Member	Description
Strength	Used to control the intensity of dehaze. The larger the value, the stronger the dehazing intensity. Value range : [0x0, 0x64] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDehazeAttr*
- *CVI_ISP_GetDehazeAttr*

24.3.2 ISP_DEHAZE_AUTO_ATTR_S

【Description】

Dehaze attribute automatic parameters

【Syntax】

```
typedef struct _ISP_DEHAZE_AUTO_ATTR_S {
 CVI_U8 Strength[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_DEHAZE_AUTO_ATTR_S;
```

【Member】

Member	Description
Strength	Used to control the intensity of dehaze. The larger the value, the stronger the dehazing intensity. Value range : [0x0, 0x64] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDehazeAttr*
- *CVI_ISP_GetDehazeAttr*

24.3.3 ISP_DEHAZE_ATTR_S

【Description】

Dehaze attribute parameters

【Syntax】

```
typedef struct _ISP_DEHAZE_ATTR_S {
 CVI_BOOL Enable; /*RW; Range:[0, 1]*/
 ISP_OP_TYPE_E enOpType;
 CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
 CVI_U16 CumulativeThr; /*RW; Range:[0x0, 0x3fff]*/
 CVI_U16 MinTransMapValue; /*RW; Range:[0x0, 0x1fff]*/
 CVI_BOOL DehazeLumaEnable; /*RW; Range:[0, 1]*/
 CVI_BOOL DehazeSkinEnable; /*RW; Range:[0, 1]*/
 CVI_U8 AirLightMixWgt; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 DehazeWgt; /*RW; Range:[0x0, 0x20]*/
 CVI_U8 TransMapScale; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 AirlightDiffWgt; /*RW; Range:[0x0, 0x10]*/
 CVI_U16 AirLightMax; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 AirLightMin; /*RW; Range:[0x0, 0xfff]*/
```

(continues on next page)

(continued from previous page)

```
  CVI_U8 SkinCb; /*RW; Range:[0x0, 0xff]*/
  CVI_U8 SkinCr; /*RW; Range:[0x0, 0xff]*/
  CVI_U16 DehazeLumaCOEFFI; /*RW; Range:[0x0, 0x7d0]*/
  CVI_U16 DehazeSkinCOEFFI; /*RW; Range:[0x0, 0x7d0]*/
  CVI_U8 TransMapWgtWgt; /*RW; Range:[0x0, 0x80]*/
  CVI_U8 TransMapWgtSigma; /*RW; Range:[0x0, 0xff]*/
  ISP_DEHAZE_MANUAL_ATTR_S stManual;
  ISP_DEHAZE_AUTO_ATTR_S stAuto;
} ISP_DEHAZE_ATTR_S;
```

【Member】

Member	Description
Enable	Enable the Dehaze function Value range: [0x0, 0x1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
CumulativeThr	Computes the statistical threshold for fog density. The default value is about 0.05% of the total number of images in the original image Value range: [0x0, 0x3fff] Data type: CVI_U16
MinTransMapValue	The minimum allowable transmission coefficient Value range: [0x0, 0x1fff] Data type: CVI_U16
DehazeLumaEnable	"Adjust the defogging intensity function according to the brightness to enable Value range: [0x0, 0x1] Data type: CVI_BOOL
DehazeSkinEnable	"Adjust the defogging intensity function according to the skin color to enable Value range: [0x0, 0x1] Data type: CVI_BOOL
AirLightMixWgt	Airlight Blend Weights Value range: [0x0, 0x20] Data type: CVI_U8
DehazeWgt	Dehaze output blend weights Value range: [0x0, 0x20] Data type: CVI_U8
TransMapScale	Transmission coefficient gain Value range: [0x0, 0xff] Data type: CVI_U8
AirlightDiffWgt	Airlight Value range: [0x0, 0x10] Data type: CVI_U8
AirLightMax	Maximum value allowed by Airlight Value range: [0x0, 0xfff] Data type: CVI_U16
AirLightMin	Minimum value allowed by Airlight Value range: [0x0, 0xfff] Data type: CVI_U16
SkinCb	The coordinates of the custom skin color on the Cb domain Value range: [0x0, 0xff] Data type: CVI_U8
SkinCr	The coordinates of the custom skin color on the Cr domain Value range: [0x0, 0xff] Data type: CVI_U8
DehazeLumaCOEFFI	According to the brightness control defogging intensity curve, the brightness is divided into 16 levels. The larger the value, the stronger the defogging strength

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDehazeAttr*
- *CVI_ISP_GetDehazeAttr*

25 ColorTone

25.1 Function Overview

On the basis of WB, further adjust the preference of red or blue image color.

25.2 API Reference

- *CVI_ISP_SetColorToneAttr* : Set color tone attribute parameter
- *CVI_ISP_GetColorToneAttr* : Get color tone attribute parameter

25.2.1 CVI_ISP_SetColorToneAttr

【Description】

Set color tone attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetColorToneAttr(VI_PIPE ViPipe, const ISP_ColorToneAttr_ATTR_S
*pstColorToneAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstColorToneAttr	Color tone attribute	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related topic】

- *CVI_ISP_GetColorToneAttr*

25.2.2 CVI_ISP_GetColorToneAttr

【Description】

Get color tone attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetColorToneAttr(VI_PIPE ViPipe, ISP_COLOR_TONE_ATTR_S␣
↪*pstWBGAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstColorToneAttr	color tone attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None

【Example】

None

【Related topic】

- *CVI_ISP_SetColorToneAttr*

25.3 Data Types

ISP_COLOR_TONE_ATTR_S: Color tone attribute parameter

25.3.1 ISP_COLOR_TONE_ATTR_S

【Description】

Color tone attribute parameter

【Syntax】

```
typedef struct _ISP_COLOR_TONE_ATTR_S {
   CVI_U16 u16RedCastGain;
   CVI_U16 u16GreenCastGain;
   CVI_U16 u16BlueCastGain;
} ISP_COLOR_TONE_ATTR_S;
```

【Member】

Member	Description
u16RedCastGain	R channel gain, 8bit decimal precision. Value range: [0x0, 0x1000] Data Type: CVI_U16
u16GreenCastGain	G channel gain, 8bit decimal precision. Value range: [0x0, 0x1000] Data Type: CVI_U16
u16BlueCastGain	B channel gain, 8bit decimal precision. Value range: [0x0, 0x1000] Data Type: CVI_U16

【Note】

None

【Related Data Type and Interface】

- *CVI_ISP_SetColorToneAttr*
- *CVI_ISP_GetColorToneAttr*

26 Saturation

26.1 Function Overview

Adjust the color saturation.

26.2 API Reference

- *CVI_ISP_SetSaturationAttr* : Set saturation attribute parameter.
- *CVI_ISP_GetSaturationAttr* : Get saturation attribute parameter.

26.2.1 CVI_ISP_SetSaturationAttr

【Description】

Set saturation attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetSaturationAttr(VI_PIPE ViPipe, const ISP_SATURATION_ATTR_S
*pstSaturationAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstSaturationAttr	Saturation attribute parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related topic】

- *CVI_ISP_GetSaturationAttr*

26.2.2 CVI_ISP_GetSaturationAttr

【Description】

Set saturation attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetSaturationAttr(VI_PIPE ViPipe, ISP_SATURATION_ATTR_S
↪*pstSaturationAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstSaturationAttr	Saturation attribute parameters	output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related topic】

- *CVI_ISP_SetSaturationAttr*

26.3 Data Types

- *ISP_SATURATION_MANUAL_ATTR_S* : Saturation attribute manual parameters
- *ISP_SATURATION_AUTO_ATTR_S* : Saturation attribute auto parameters
- *ISP_SATURATION_ATTR_S* : Saturation attribute parameters

26.3.1 ISP_SATURATION_MANUAL_ATTR_S

【Description】

Saturation attribute manual parameters

【Syntax】

```
typedef struct _ISP_SATURATION_MANUAL_ATTR_S {
    CVI_U8 Saturation;
} ISP_SATURATION_MANUAL_ATTR_S;
```

【Member】

Member	Description
Saturation	An array of three values defining the UV gain for the output. Find the UV gain according to the input saturation. The larger the value, the higher the saturation; otherwise, the smaller. Value range: [0x0, 0xFF] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSaturationAttr*
- *CVI_ISP_GetSaturationAttr*

26.3.2 ISP_SATURATION_AUTO_ATTR_S

【Description】

Saturation attribute auto parameters

【Syntax】

```
typedef struct _ISP_SATURATION_AUTO_ATTR_S {
    CVI_U8 Saturation[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_SATURATION_AUTO_ATTR_S;
```

【Member】

Member	Description
Saturation	An array of three values defining the UV gain for the output. Find the UV gain according to the input saturation. The larger the value, the higher the saturation; otherwise, the smaller. Value range: [0x0, 0xFF] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSaturationAttr*
- *CVI_ISP_GetSaturationAttr*

26.3.3 ISP_SATURATION_ATTR_S

【Description】

Saturation attribute parameters

【Syntax】

```
typedef struct _ISP_SATURATION_ATTR_S {
   ISP_OP_TYPE_E enOpType;
   ISP_SATURATION_AUTO_ATTR_S stAuto;
   ISP_SATURATION_MANUAL_ATTR_S stManual;
} ISP_SATURATION_ATTR_S;
```

【Member】

Member	Description
enOpType	Working type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
stAuto	Manual mode parameter attributes
stManual	Automatic mode parameter attributes

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetSaturationAttr*
- *CVI_ISP_GetSaturationAttr*

27 CAC

27.1 Function Overview

Color distortion correction: Due to the different light refractive index of different colors, purple edges are easy to form around the objects at the junction of the highlighted area and the low-lit area. This module can realize the function of removing purple edges in the image and improve the purple edge phenomenon in the image edge.

27.2 API Reference

- *CVI_ISP_SetCacAttr*: Set gamma attribute parameters
- *CVI_ISP_GetCacAttr*: Get gamma attribute parameters

27.2.1 CVI_ISP_SetCacAttr

【Description】

Set gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetCacAttr(VI_PIPE ViPipe, const ISP_CacAttr_ATTR_S
↪*pstCacAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCacAttr	Gamma attribute parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCacAttr*

27.2.2 CVI_ISP_GetCacAttr

【Description】

Get gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetCacAttr(VI_PIPE ViPipe, ISP_CacAttr_ATTR_S *pstCacAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCacAttr	Gamma attribute parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCacAttr*

27.3 Data Types

- *ISP_CAC_MANUAL_ATTR_S* : Gamma attribute manual parameter
- *ISP_CAC_AUTO_ATTR_S* : Gamma attribute automatic parameter
- *ISP_CAC_ATTR_S* : Gamma attribute parameter

27.3.1 ISP_CAC_MANUAL_ATTR_S

【Description】

Gamma attribute manual parameter

【Syntax】

```
typedef struct _ISP_CAC_MANUAL_ATTR_S {
 CVI_U8 DePurpleStr; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 EdgeGlobalGain; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 EdgeCoring; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 EdgeStrMin; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 EdgeStrMax; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 DePurpleCbStr; /*RW; Range:[0x0, 0x8]*/
 CVI_U8 DePurpleCrStr; /*RW; Range:[0x0, 0x8]*/
 CVI_U8 DePurpleStrMaxRatio; /*RW; Range:[0x0, 0x40]*/
 CVI_U8 DePurpleStrMinRatio; /*RW; Range:[0x0, 0x40]*/
} ISP_CAC_MANUAL_ATTR_S;
```

【Member】

Member	Description
DePurpleStr	Depurple fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
EdgeGlobalGain	Intensity total gain for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoring	Noise Suppression Control for Edge Detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeStrMin	Intensity lower limit for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeStrMax	Strength limit for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleCbStr	B channel purple fringing corrected edge intensity. The larger the value, the less purple fringing Value range: [0x0, 0x8] Data type: CVI_U8
DePurpleCrStr	R channel purple fringing corrected edge intensity. The larger the value, the less purple fringing Value range: [0x0, 0x8] Data type: CVI_U8
DePurpleStrMaxRatio	The maximum allowable gain for defrringing intensity Value range: [0x0, 0x40] Data type: CVI_U8
DePurpleStrMinRatio	Minimum allowable gain for defrringing intensity Value range: [0x0, 0x40] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCacAttr*
- *CVI_ISP_GetCacAttr*

27.3.2 ISP_CAC_AUTO_ATTR_S

【Description】

Gamma attribute automatic parameter

【Syntax】

```
typedef struct _ISP_CAC_AUTO_ATTR_S {
    CVI_U8 DePurpleStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeGlobalGain[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeStrMin[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 EdgeStrMax[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCbStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleCrStr[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleStrMaxRatio[ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U8 DePurpleStrMinRatio[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CAC_AUTO_ATTR_S;
```

【Member】

Member	Description
DePurpleStr	Depurple fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
EdgeGlobalGain	Intensity total gain for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoring	Noise Suppression Control for Edge Detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeStrMin	Intensity lower limit for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeStrMax	Strength limit for edge detection Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleCbStr	B channel purple fringing corrected edge intensity. The larger the value, the less purple fringing Value range: [0x0, 0x8] Data type: CVI_U8
DePurpleCrStr	R channel purple fringing corrected edge intensity. The larger the value, the less purple fringing Value range: [0x0, 0x8] Data type: CVI_U8
DePurpleStrMaxRatio	The maximum allowable gain for defrringing intensity Value range: [0x0, 0x40] Data type: CVI_U8
DePurpleStrMinRatio	Minimum allowable gain for defrringing intensity Value range: [0x0, 0x40] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCacAttr*
- *CVI_ISP_GetCacAttr*

27.3.3 ISP_CAC_ATTR_S

【Description】

Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_CAC_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_U8 PurpleDetRange;
  CVI_U8 PurpleCb;
  CVI_U8 PurpleCr;
  CVI_U8 PurpleCb2;
  CVI_U8 PurpleCr2;
  CVI_U8 PurpleCb3;
  CVI_U8 PurpleCr3;
  CVI_U8 GreenCb;
  CVI_U8 GreenCr;
  CVI_U8 TuningMode;
  CVI_U8 EdgeGainIn[3];
  CVI_U8 EdgeGainOut[3];
  ISP_CAC_MANUAL_ATTR_S stManual;
  ISP_CAC_AUTO_ATTR_S stAuto;
} ISP_CAC_ATTR_S;
```

【Member】

Member	Description
Enable	CAC module enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
PurpleDetRange	Threshold for purple fringing detection. The larger the value, the more areas are judged as purple fringing Value range: [0x0, 0x80] Data type: CVI_U8
PurpleCb	The coordinates of purple in Cb domain Value range: [0x0, 0xff] Data type: CVI_U8
PurpleCr	The coordinates of purple in Cr domain Value range: [0x0, 0xff] Data type: CVI_U8
PurpleCb2	The coordinates of purple 2 in Cb domain Value range: [0x0, 0xff] Data type: CVI_U8
PurpleCr2	The coordinates of purple 2 in the Cr domain Value range: [0x0, 0xff] Data type: CVI_U8
PurpleCb3	The coordinates of purple 3 in the Cb domain. Value range: [0x0, 0xff] Data type: CVI_U8
PurpleCr3	The coordinates of purple 3 in the Cr domain Value range: [0x0, 0xff] Data type: CVI_U8
GreenCb	Coordinates of green in Cb domain Value range: [0x0, 0xff] Data type: CVI_U8
GreenCr	Coordinates of green in Cr domain Value range: [0x0, 0xff] Data type: CVI_U8
TuningMode	Output debug strategy to assist in adjusting parameters. 0: CAC image result. 1: Edge detection image result. 2: The value range of purple fringe detection image results: [0x0, 0x2] Data type: CVI_U8
EdgeGainIn[3]	An array of four numeric values. Define the edge strength level, the larger the value, the stronger the edge strength
	Value range: [0x0, 0x20] Data type: CVI_U8
EdgeGainOut[3]	An array of four numeric values. Define the

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCacAttr*
- *CVI_ISP_GetCacAttr*

28 LCAC

28.1 Function Overview

Color distortion correction. Due to the different refractive index of different colors, purple fringing is easy to form around objects at the junction of high-brightness areas and low-brightness areas. This module realizes the image de-purple fringing function and improves the purple fringing phenomenon at the edge of the image.

28.2 API Reference

- *CVI_ISP_SetLCACAttr* : Set gamma attribute parameters
- *CVI_ISP_GetLCACAttr* : Get gamma attribute parameters

28.2.1 CVI_ISP_SetLCACAttr

【Description】

Set gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetLCACAttr(VI_PIPE ViPipe, const ISP_LCAC_ATTR_S *pstLCACAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstLCACAttr	Gamma property parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related topic】

- *CVI_ISP_GetLCACAttr*

28.2.2 CVI_ISP_GetLCACAttr

【Description】

Get gamma attribute parameters

【Syntax】

```

```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstLCACAttr	Gamma property parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related topic】

- *CVI_ISP_SetLCACAttr*

28.3 Data Types

- *ISP_LCAC_GAUSS_COEF_ATTR_S* : Gamma Gauss parameter
- *ISP_LCAC_MANUAL_ATTR_S* : Gamma attribute manual parameter
- *ISP_LCAC_AUTO_ATTR_S* : Gamma attribute automatic parameter
- *ISP_LCAC_ATTR_S* : Gamma attribute parameter

28.3.1 ISP_LCAC_GAUSS_COEF_ATTR_S

【Description】

Gamma Gauss parameter

【Syntax】

```
typedef struct _ISP_LCAC_GAUSS_COEF_ATTR_S {
  CVI_U8 Wgt;
  CVI_U8 Sigma;
} ISP_LCAC_GAUSS_COEF_ATTR_S;
```

【Member】

Member	Description
Wgt	Control the degree of purple fringing according to the edge strength, and determine the weight curve of the fusion of the purple fringing result and the original image. The larger the Wgt, the larger the ratio of the fusion original image Value range: [0x0, 0x80] Data type: CVI_U8
Sigma	Control the degree of purple fringing according to the edge strength, and determine the weight curve of the fusion of the purple fringing result and the original image. The larger the Sigma, the more pixels are fused with the original image Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】 None.

28.3.2 ISP_LCAC_MANUAL_ATTR_S

【Description】

Gamma attribute manual parameter

【Syntax】

```
typedef struct _ISP_LCAC_MANUAL_ATTR_S {
 CVI_U16 DePurpleCrGain;
 CVI_U16 DePurpleCbGain;
 CVI_U8 DePurepleCrWgt0;
 CVI_U8 DePurepleCbWgt0;
 CVI_U8 DePurepleCrWgt1;
 CVI_U8 DePurepleCbWgt1;
 CVI_U8 EdgeCoringBase;
 CVI_U8 EdgeCoringAdv;
} ISP_LCAC_MANUAL_ATTR_S;
```

【Member】

Member	Description
DePurpleCrGain	R channel purple fringing correction gain Value range: [0x0, 0xfff] Data type: CVI_U16
DePurpleCbGain	B channel purple fringing correction gain Value range: [0x0, 0xfff] Data type: CVI_U16
DePurepleCrWgt0	Long Frame R Channel Corrected Intensity Blend Weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCbWgt0	Long Frame B Channel Corrected Intensity Blend Weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCrWgt1	Short frame R channel corrected intensity blend weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCbWgt1	Short frame B channel corrected intensity blend weights Value range: [0x0, 0x40] Data type: CVI_U8
EdgeCoringBase	Noise Suppression Control for Base Edge Detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoringAdv	Noise suppression control for Advance edge detection Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLCACAttr*
- *CVI_ISP_GetLCACAttr*

28.3.3 ISP_LCAC_AUTO_ATTR_S

【Description】

Gamma attribute automatic parameter

【Syntax】

```
typedef struct _ISP_LCAC_AUTO_ATTR_S {
  CVI_U16 DePurpleCrGain[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 DePurpleCbGain[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DePurepleCrWgt0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DePurepleCbWgt0[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DePurepleCrWgt1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 DePurepleCbWgt1[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 EdgeCoringBase[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U8 EdgeCoringAdv[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_LCAC_AUTO_ATTR_S;
```

【Member】

Member	Description
DePurpleCrGain	R channel purple fringing correction gain Value range: [0x0, 0xfff] Data type: CVI_U16
DePurpleCbGain	B channel purple fringing correction gain Value range: [0x0, 0xfff] Data type: CVI_U16
DePurepleCrWgt0	Long Frame R Channel Corrected Intensity Blend Weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCbWgt0	Long Frame B Channel Corrected Intensity Blend Weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCrWgt1	Short frame R channel corrected intensity blend weights Value range: [0x0, 0x40] Data type: CVI_U8
DePurepleCbWgt1	Short frame B channel corrected intensity blend weights Value range: [0x0, 0x40] Data type: CVI_U8
EdgeCoringBase	Noise Suppression Control for Base Edge Detection Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoringAdv	Noise suppression control for Advance edge detection Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLCACAttr*
- *CVI_ISP_GetLCACAttr*

28.3.4 ISP_LCAC_ATTR_S

【Description】

Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_LCAC_ATTR_S {
 CVI_BOOL Enable;
 ISP_OP_TYPE_E enOpType;
 CVI_U8 UpdateInterval;
 CVI_U8 TuningMode;
 CVI_U8 DePurpleCrStr0;
 CVI_U8 DePurpleCbStr0;
 CVI_U8 DePurpleCrStr1;
 CVI_U8 DePurpleCbStr1;
 CVI_U8 FilterTypeBase;
 CVI_U8 EdgeGainBase0;
 CVI_U8 EdgeGainBase1;
 CVI_U8 EdgeStrWgtBase;
 CVI_U8 DePurpleStrMaxBase;
 CVI_U8 DePurpleStrMinBase
 CVI_U8 FilterScaleAdv;
 CVI_U8 LumaWgt;
 CVI_U8 FilterTypeAdv;
 CVI_U8 EdgeGainAdv0;
 CVI_U8 EdgeGainAdv1;
 CVI_U8 EdgeStrWgtAdvG;
 CVI_U8 DePurpleStrMaxAdv;
 CVI_U8 DePurpleStrMinAdv;
 ISP_LCAC_GAUSS_COEF_ATTR_S EdgeWgtBase;
 ISP_LCAC_GAUSS_COEF_ATTR_S EdgeWgtAdv;
 ISP_LCAC_MANUAL_ATTR_S stManual;
 ISP_LCAC_AUTO_ATTR_S stAuto;
} ISP_LCAC_ATTR_S;
```

【Member】

Member	Description
Enable	LCAC module enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
TuningMode	Threshold for edge detection. The smaller the value, the more regions are judged as edges Value range: [0x0, 0x6] Data type: CVI_U8
DePurpleCrStr0	Long frame R channel correction strength Value range: [0x0, 0x40] Data type: CVI_U8
DePurpleCbStr0	Long Frame B Channel Correction Intensity Value range: [0x0, 0x40] Data type: CVI_U8
DePurpleCrStr1	Short frame R channel correction strength Value range: [0x0, 0x40] Data type: CVI_U8
DePurpleCbStr1	Short frame B channel correction strength Value range: [0x0, 0x40] Data type: CVI_U8
FilterTypeBase	Filter selection. The larger the value, the stronger the purple fringing strength Value range: [0x0, 0x3] Data type: CVI_U8
EdgeGainBase0	Intensity gain for long frame edge detection Value range: [0x0, 0x1c] Data type: CVI_U8
EdgeGainBase1	Intensity Gain for Short Frame Edge Detection Value range: [0x0, 0x23] Data type: CVI_U8
EdgeStrWgtBase	Base channel edge detection weight Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleStrMaxBase	Base Upper limit for de-fringing intensity Value range: [0x0, 0x80] Data type: CVI_U8
DePurpleStrMinBase	Base Lower limit for de-fringing intensity Value range: [0x0, 0x80] Data type: CVI_U8
FilterScaleAdv	filter scale parameter Value range: [0x0, 0xf] Data type: CVI_U8
LumaWgt	Luma Blend Weight Value range: [0x0, 0x1ff]
	Data type: CVI_U16
FilterTypeAdv	filter selection. The larger the value, the stronger the purple fringing strength

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetLCACAttr*
- *CVI_ISP_GetLCACAttr*

29 RGBCAC

29.1 Function Overview

- To achieve the image to remove purple edge function, improve the image edge purple phenomenon.

29.2 API Reference

- *CVI_ISP_SetRGBCACAttr* : Set gamma attribute parameters
- *CVI_ISP_GetRGBCACAttr* : Get gamma attribute parameters

29.2.1 CVI_ISP_SetRGBCACAttr

【Description】

Set gamma attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetRGBCACAttr(VI_PIPE ViPipe, const ISP_RGBCAC_ATTR_S␣
↪*pstRGBCACAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstRGBCACAttr	Gamma property parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetRGBCACAttr*

29.2.2 CVI_ISP_GetRGBCACAttr

【Description】

Get gamma attribute parameters

【Syntax】

```
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstRGBCACAttr	Gamma property parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetRGBCACAttr*

29.3 Data Types

- *ISP_RGBCAC_MANUAL_ATTR_S* : Gamma attribute manual parameter
- *ISP_RGBCAC_AUTO_ATTR_S* : Gamma attribute automatic parameter
- *ISP_RGBCAC_ATTR_S* : Gamma attribute parameter

29.3.1 ISP_RGBCAC_MANUAL_ATTR_S

【Description】

Gamma attribute manual parameter

【Syntax】

```
typedef struct _ISP_RGBCAC_MANUAL_ATTR_S {
 CVI_U8 DePurpleStr0; /*RW; Range:[0x0, 0xFF]*/
 CVI_U8 DePurpleStr1; /*RW; Range:[0x0, 0xFF]*/
 CVI_U16 EdgeCoring; /*RW; Range:[0x0, 0xFFF]*/
 CVI_U8 DePurpleCrStr0; /*RW; Range:[0x0, 0x10]*/
 CVI_U8 DePurpleCbStr0; /*RW; Range:[0x0, 0x10]*/
 CVI_U8 DePurpleCrStr1; /*RW; Range:[0x0, 0x10]*/
 CVI_U8 DePurpleCbStr1; /*RW; Range:[0x0, 0x10]*/
} ISP_RGBCAC_MANUAL_ATTR_S;
```

【Member】

Member	Description
DePurpleStr0	Long frame de-fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleStr1	Short frame de-fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoring	Noise Suppression Control for Edge Detection Value range: [0x0, 0xfff] Data type: CVI_U16
DePurpleCrStr0	Long frame R channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCbStr0	Long frame B channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCrStr1	Short frame R channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCbStr1	Short frame B channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRGBCACAttr*
- *CVI_ISP_GetRGBCACAttr*

29.3.2 ISP_RGBCAC_AUTO_ATTR_S

【Description】

Gamma attribute automatic parameter

【Syntax】

```
typedef struct _ISP_RGBCAC_AUTO_ATTR_S {
 CVI_U8 DePurpleStr0[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DePurpleStr1[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U16 EdgeCoring[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DePurpleCrStr0[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DePurpleCbStr0[ISP_AUTO_ISO_STRENGTH_NUM];
 CVI_U8 DePurpleCrStr1[ISP_AUTO_ISO_STRENGTH_NUM];
```

(continues on next page)

(continued from previous page)

```
  CVI_U8 DePurpleCbStr1[ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_RGBCAC_AUTO_ATTR_S;
```

【Member】

Member	Description
DePurpleStr0	Long frame de-fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleStr1	Short frame de-fringing intensity. The larger the value, the less purple fringing Value range: [0x0, 0xff] Data type: CVI_U8
EdgeCoring	Noise Suppression Control for Edge Detection Value range: [0x0, 0xfff] Data type: CVI_U16
DePurpleCrStr0	Long frame R channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCbStr0	Long frame B channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCrStr1	Short frame R channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8
DePurpleCbStr1	Short frame B channel corrected edge intensity Value range: [0x0, 0x10] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRGBCACAttr*
- *CVI_ISP_GetRGBCACAttr*

29.3.3 ISP_RGBCAC_ATTR_S

【Description】

Gamma attribute parameter

【Syntax】

```
typedef struct _ISP_RGBCAC_ATTR_S {
  CVI_BOOL Enable;
```

(continues on next page)

(continued from previous page)

```
  ISP_OP_TYPE_E enOpType;
  CVI_U16 VarThr;
  CVI_U8 PurpleDetRange0;
  CVI_U8 PurpleDetRange1;
  CVI_U8 DePurpleStrMax0;
  CVI_U8 DePurpleStrMin0;
  CVI_U8 DePurpleStrMax1;
  CVI_U8 DePurpleStrMin1;
  CVI_U16 EdgeGlobalGain;
  CVI_U8 EdgeGainIn[3];
  CVI_U8 EdgeGainOut[3];
  CVI_U16 LumaScale;
  CVI_U16 UserDefineLuma;
  CVI_U8 LumaBlendWgt;
  CVI_U8 LumaBlendWgt2;
  CVI_U8 LumaBlendWgt3;
  CVI_U8 PurpleCb;
  CVI_U8 PurpleCr;
  CVI_U8 PurpleCb2;
  CVI_U8 PurpleCr2;
  CVI_U8 PurpleCb3;
  CVI_U8 PurpleCr3;
  CVI_U8 GreenCb;
  CVI_U8 GreenCr;
  CVI_U8 TuningMode;
  ISP_RGBCAC_MANUAL_ATTR_S stManual;
  ISP_RGBCAC_AUTO_ATTR_S stAuto;
} ISP_RGBCAC_ATTR_S;
```

【Member】

Member	Description
Enable	CAC module enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
VarThr	Threshold for edge detection. The smaller the value, the more regions are judged as edges Value range: [0x0, 0x1ff] Data type: CVI_U16
PurpleDetRange0	Threshold for long frame purple fringe detection. The larger the value, the more areas are judged as purple fringing Value range: [0x0, 0x80] Data type: CVI_U8
PurpleDetRange1	Threshold for short frame purple fringing detection. The larger the value, the more areas are judged as purple fringing Value range: [0x0, 0x80] Data type: CVI_U8
DePurpleStrMax0	Allowable maximum value of long frame depurple intensity Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleStrMin0	The allowable minimum value of the long frame depurple intensity Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleStrMax1	Allowable maximum value of short frame depurple intensity Value range: [0x0, 0xff] Data type: CVI_U8
DePurpleStrMin1	The minimum value allowed for the intensity of purple fringing in short frames Value range: [0x0, 0xff] Data type: CVI_U8
EdgeGlobalGain	Intensity total gain for edge detection Value range: [0x0, 0xfff] Data type: CVI_U16
EdgeGainIn[3]	An array of four numeric values. Define the edge strength level, the larger the value, the stronger the edge strength Value range: [0x0, 0x3] Data type: CVI_U8
EdgeGainOut[3]	An array of four numeric values. Define the edge strength gain, the larger the value, the stronger the purple fringing effect Value range: [0x0, 0x3] Data type: CVI_U8
LumaScale	brightness control Value range: [0x0, 0x7ff] Data type: CVI_U16
UserDefineLuma	custom brightness value Value range: [0x0, 0xfff] Data type: CVI_U16
LumaBlendWgt	Blend weights for luma 1 Value range: [0x0, 0x20] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetRGBCACAttr*
- *CVI_ISP_GetRGBCACAttr*

30 FSWDR

30.1 Function Overview

Wide dynamic range synthesis of related parameters.

30.2 API Reference

- *CVI_ISP_SetFSWDRAttr* : Set frame composition attribute parameters
- *CVI_ISP_GetFSWDRAttr* : Get frame composition attribute parameters
- *CVI_ISP_SetWDRExposureAttr* : Set WDR exposure attribute parameters
- *CVI_ISP_GetWDRExposureAttr* : Get WDR exposure attribute parameters

30.2.1 CVI_ISP_SetFSWDRAttr

【Description】

Set frame composition attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetFSWDRAttr(VI_PIPE ViPipe, const ISP_FSWDR_ATTR_S
*pstFSWDRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstFSWDRAttr	Frame composition attribute parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- cv180x does not support this function
- It is recommended that Gamma table be set to sRGB when wdr mode is enabled

【Example】

None.

【Related topic】

- *CVI_ISP_GetFSWDRAttr*

30.2.2 CVI_ISP_GetFSWDRAttr

【Description】

Get frame composition attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetFSWDRAttr(VI_PIPE ViPipe, ISP_FSWDR_ATTR_S *pstFSWDRAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstFSWDRAttr	Frame composition attribute parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- cv180x does not support this function

【Example】

- It is recommended that Gamma table be set to sRGB when wdr mode is enabled

【Related topic】

- *CVI_ISP_SetFSWDRAttr*

30.2.3 CVI_ISP_SetWDRExposureAttr

【Description】

Set WDR exposure attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_SetWDRExposureAttr(VI_PIPE ViPipe, const ISP_WDR_EXPOSURE_ATTR_
↪S *pstWDRExpAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstWDRExposure-Attr	WDR exposure attribute parameters	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- cv180x does not support this function

【Example】

- It is recommended that Gamma table be set to sRGB when wdr mode is enabled

【Related topic】

- *CVI_ISP_GetWDRExposureAttr*

30.2.4 CVI_ISP_GetWDRExposureAttr

【Description】

Get WDR exposure attribute parameters

【Syntax】

```
CVI_S32 CVI_ISP_GetWDRExposureAttr(VI_PIPE ViPipe, ISP_WDRExposureAttr_ATTR_S␣
↪*pstWDRExposureAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstWDRExposure-Attr	WDR exposure attribute parameters	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- cv180x does not support this function
- It is recommended that Gamma table be set to sRGB when wdr mode is enabled

【Example】

None.

【Related topic】

- *CVI_ISP_SetWDRExposureAttr*

30.3 Data Types

- *ISP_FSWDR_MANUAL_ATTR_S* : Frame composition attribute manual parameters
- *ISP_FSWDR_AUTO_ATTR_S* : Frame composition attribute automatic parameters
- *ISP_FSWDR_ATTR_S* : Frame composition attribute parameters
- *ISP_WDR_EXPOSURE_ATTR_S* : WDR exposure attribute parameters

30.3.1 ISP_FSWDR_MANUAL_ATTR_S

【Description】

Frame composition attribute manual parameters

【Syntax】

```
typedef struct _ISP_FSWDR_MANUAL_ATTR_S {
 CVI_U16 WDRCombineLongThr; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRCombineShortThr; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRCombineMaxWeight; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRCombineMinWeight; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 WDRMtIn[4]; /*RW; Range:[0x0, 0xff]*/
 CVI_U16 WDRMtOut[4]; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRLongWgt; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 WDRCombineSNRAwareToleranceLevel; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 MergeModeAlpha; /*RW; Range:[0x0, 0xff]*/
 CVI_U16 WDRMotionCombineLongThr; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRMotionCombineShortThr; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRMotionCombineMinWeight; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRMotionCombineMaxWeight; /*RW; Range:[0x0, 0x100]*/
} ISP_FSWDR_MANUAL_ATTR_S;
```

【Member】

Member	Description
WDRCombineLongThr	Long exposure critical value, image data below this critical value will only select long exposure data to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRCombineShortThr	Short exposure threshold, if the image data exceeds this threshold, only short exposure data will be selected to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRCombineMinWeight	The minimum weight value for long and short exposure image data fusion. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. (This value must be less than or equal to WDRCombineMaxWeight) Value range: [0x0, 0x100] Data type: CVI_U8
WDRCombineMaxWeight	Long and short exposure image data are fused with the highest weight value. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. (This value must be greater than or equal to WDRCombineMinWeight) Value range: [0x0, 0x100] Data type: CVI_U8
WDRMotionCombineLongThr	Motion detection information, long exposure threshold, image data below this threshold will only select long exposure data to synthesize a WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRMotionCombineShortThr	Motion detection information, short exposure critical value, image data exceeding this critical value will only select short exposure data to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRMotionCombineMinWeight	Motion detection information, long and short exposure image data fusion minimum weight value. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. Value range: [0x0, 0x100] Data type: CVI_U16
WDRMotionCombineMaxWeight	Motion detection information, long and short exposure image data fusion with the highest weight value. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. Value range: [0x0, 0x100] Data type: CVI_U16
WDRMtIn[4]	An array of four numeric values. Define the amount of mo-
	tion of the input object, the larger the value, the greater the amount of motion. Value range: [0x0, 0xff]

【Note】

- The preceding parameter cv180x is not supported

【Related Data Type and Interface】

- *CVI_ISP_SetFSWDRAttr*
- *CVI_ISP_GetFSWDRAttr*

30.3.2 ISP_FSWDR_AUTO_ATTR_S

【Description】

Frame composition attribute automatic parameters

【Syntax】

```
typedef struct cviISP_FSWDRAttr_AUTO_ATTR_S {
 CVI_U16 WDRCombineLongThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRCombineShortThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRCombineMaxWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRCombineMinWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 WDRMtIn[4][ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U16 WDRMtOut[4][ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRLongWgt[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 WDRCombineSNRAwareToleranceLevel[ISP_AUTO_LV_NUM];
 CVI_U8 MergeModeAlpha[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/
 CVI_U16 WDRMotionCombineLongThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRMotionCombineShortThr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xfff]*/
 CVI_U16 WDRMotionCombineMinWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
 CVI_U16 WDRMotionCombineMaxWeight[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
}ISP_FSWDR_AUTO_ATTR_S;
```

【Member】

Member	Description
WDRCombineLongThr	Long exposure critical value, image data below this critical value will only select long exposure data to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRCombineMinWeight	The minimum weight value for long and short exposure image data fusion. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. (This value must be less than or equal to WDRCombineMaxWeight) Value range: [0x0, 0xff] Data type: CVI_U8
WDRMotionCombine-LongThr	Motion detection information, long exposure threshold, image data below this threshold will only select long exposure data to synthesize a WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRMotionCombineShort-Thr	Motion detection information, short exposure critical value, image data exceeding this critical value will only select short exposure data to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRMotionCombineM-inWeight	Motion detection information, long and short exposure image data fusion minimum weight value. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. Value range: [0x0, 0x100] Data type: CVI_U16
WDRMotionCombine-MaxWeight	Motion detection information, long and short exposure image data fusion with the highest weight value. The larger the weight value, the more the proportion of long exposure will be in the fusion time, and vice versa, the more proportion of short exposure will be. Value range: [0x0, 0x100] Data type: CVI_U16
MergeModeAlpha	The ratio of mobile information fusion when MergeMode is set to 1. The larger the value is, the more the proportion of short frame motion information is in the fusion. On the contrary, the proportion of long frame motion information is more. Value range: [0x0, 0xff] Data type: CVI_U8
WDRCombineSNRAware-ToleranceLevel	Noise Tolerance Strength of Short Frame Noise SNR Adaptive Value range: [0x0, 0xBB8] Data type: CVI_U16
WDRCombineLongThr	Long exposure critical value, image data below this critical value will only select long exposure data to synthesize WDR image. Value range: [0x0, 0xfff] Data type: CVI_U16
WDRCombineShortThr	Short exposure threshold, if the image data exceeds this threshold, only short exposure data will be selected to synthesize WDR image.

【Note】

- The preceding parameter cv180x is not supported

【Related Data Type and Interface】

- *CVI_ISP_SetFSWDRAttr*
- *CVI_ISP_GetFSWDRAttr*

30.3.3 ISP_FSWDR_ATTR_S

【Description】

Frame composition attribute parameters

【Syntax】

```
typedef struct _ISP_FSWDR_ATTR_S {
  CVI_BOOL Enable; /*RW; Range:[0, 1]*/
  ISP_OP_TYPE_E enOpType; /*RW; Range:[0, 1]*/
  CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
  CVI_BOOL MotionCompEnable; /*RW; Range:[0, 1]*/
  CVI_U8 TuningMode; /*RW; Range:[0x0, 0x9]*/
  CVI_BOOL WDRDCMode;; /*RW; Range:[0, 1]*/
  CVI_BOOL WDRLumaMode; /*RW; Range:[0x0, 0x1]*/
  CVI_U8 WDRType; /*RW; Range:[0x0, 0x2]*/
  CVI_BOOL WDRCombineSNRAwareEn; /*RW; Range:[0, 1]*/
  CVI_U16 WDRCombineSNRAwareLowThr; /*RW; Range:[0x0, 0xffff]*/
  CVI_U16 WDRCombineSNRAwareHighThr; /*RW; Range:[0x0, 0xffff]*/
  CVI_U16 WDRCombineSNRAwareSmoothLevel; /*RW; Range:[0x0, 0xbb8]*/
  CVI_BOOL LocalToneRefinedDCMode; /*RW; Range:[0, 1]*/
  CVI_BOOL LocalToneRefinedLumaMode; /*RW; Range:[0, 1]*/
  CVI_U16 DarkToneRefinedThrL; /*RW; Range:[0x0, 0xfff]*/
  CVI_U16 DarkToneRefinedThrH; /*RW; Range:[0x0, 0xfff]*/
  CVI_U16 DarkToneRefinedMaxWeight; /*RW; Range:[0x0, 0x100]*/
  CVI_U16 DarkToneRefinedMinWeight; /*RW; Range:[0x0, 0x100]*/
  CVI_U16 BrightToneRefinedThrL; /*RW; Range:[0x0, 0xfff]*/
  CVI_U16 BrightToneRefinedThrH; /*RW; Range:[0x0, 0xfff]*/
  CVI_U16 BrightToneRefinedMaxWeight; /*RW; Range:[0x0, 0x100]*/
  CVI_U16 BrightToneRefinedMinWeight; /*RW; Range:[0x0, 0x100]*/
  CVI_U8 WDRMotionFusionMode; /*RW; Range:[0x0, 0x3]*/
  CVI_BOOL MtMode; /*RW; Range:[0, 1]*/

  ISP_FSWDR_MANUAL_ATTR_S stManual;
  ISP_FSWDR_AUTO_ATTR_S stAuto;
} ISP_FSWDR_ATTR_S;
```

【Member】

Member	Description
Enable	The WDR module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
UpdateInterval	Affects the parameter update terval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
MotionCompEnable	WDR motion detection enable switch. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
WDRDCMode	Long-short exposure fusion mode 0: normal mode, adjusted by WDRLumaMode 1: DC mode Value range: [0, 1] Data type: CVI_BOOL
WDRLumaMode	Brightness Calculation Mode 0: Take the maximum value of R/G/B. 1: Take the Luma value. Value range: [0, 1] Data type: CVI_BOOL
WDRType	Custom WDR fusion mode. 0: Output the short exposure value multiplied by the exposure ratio. 1: Output long exposure value. 2: Output the value after long-short exposure fusion. The weight of long exposure is WDRLongWgt. It takes effect only when MotionCompEnable is enabled. Value range: [0, 2] Data type: CVI_U8
WDRDitherEnable	Enable the dither function for the long exposure threshold. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_U8
WDRDitherBit	Set how many bits of dither to generate. Value range: [0, 0xf] Data type: CVI_U8
WDRDitherRange	Sets the range between long and short exposure thresholds plus dither. Value range: [0, 0xfff] Data type: CVI_U16
TuningMode	Debug mode, output visual auxiliary information to help users debug 0: no visual auxiliary information 1: output short frame information in SDR form. 2: Output short frame information in HDR form. 3: Output
	long frame information in SDR form. Value range: [0x0, 0x3] Data type: CVI_U8

【Note】

- The preceding parameter cv180x is not supported

【Related Data Type and Interface】

- *CVI_ISP_SetFSWDRAttr*
- *CVI_ISP_GetFSWDRAttr*

30.3.4 ISP_WDR_EXPOSURE_ATTR_S

【Description】

WDR exposure attribute parameters

【Syntax】

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

【Member】

Member	Description
enExpRatioType	Description
au32ExpRatio[WDR_EXP_RATIO_NUM]	Only valid in multi frame synthesis WDR mode. au32ExpRatio is invalid when enExpRatioType is OP_TYPE_AUTO. When enExpRatioType is OP_TYPE_MANUAL, au32ExpRatio is readable and writable, which indicates the expected exposure ratio of two adjacent frames of multi frame synthetic WDR. Value range:[0x40, 0xFFF]
u32ExpRatioMax	Only valid in multi frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMax is the maximum value of the exposure time ratio of the longest frame to the shortest frame. When enExpRatioType is OP_TYPE_MANUAL, u32ExpRatioMax is invalid. 6bit decimal precision, 0x40 means that the exposure ratio is 1 times.
u32ExpRatioMin	Only valid in multi frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMax is the minimum value of the exposure time ratio of the longest frame to the shortest frame. When enExpRatioType is OP_TYPE_MANUAL, u32ExpRatioMax is invalid. The format is 6.6bit fixed-point without sign, and 0x40 means that the ratio of long frame exposure time to short frame exposure time is one time. The default value is 0x40. Value range:[0x40, u32ExpRatioMax]
u16Tolerance	Description
u16Speed	The auto exposure ratio adjustment speed. This is only effective in two frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the higher the value is, the faster the adjustment speed is. The default value is 0x20. Value range:[0x0, 0xFF]
u16RatioBias	The exposure ratio deviation value which is only valid in multi frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the higher the value, the greater the auto exposure ratio. The default value is 0x400, which means that the calculation result of the auto exposure ratio algorithm will not be adjusted. The exposure ratio adjusted by this value is limited by the maximum / minimum exposure ratio. Value range:[0x0, 0xFFFF]
u8SECompensation	Adjust the target brightness value of short detection screen Value range:[0x0, 0xFF]
u16SEHisThr	Calculate the threshold value of exposure ratio when the short detection frame exceeds the long detection frame (1x = 64)
u16SEHisCntRatio1	If the number of short detection histogram bin 255 is greater than this threshold, the target brightness of short detection will be reduced
u16SEHisCntRatio2	If the number of short detection histogram bin 255 is less than this threshold, the target brightness of the reduced short detection will return to the original target brightness
u16SEHis255CntThr1	If the number of long detection histogram bin 255 is greater than this threshold, the target brightness of the long detection
	will be reduced
u16SEHis255CntThr2	If the number of long detection histogram bin 255 is less than this threshold, the target brightness of the reduced long de-

【Note】

- The preceding parameter cv180x is not supported

【Related Data Type and Interface】

- *CVI_ISP_SetWDRExposureAttr*
- *CVI_ISP_GetWDRExposureAttr*

31 DRC

31.1 Function Overview

The dynamic range of the whole image makes the display effect on the display device consistent with the human visual experience.

31.2 API Reference

- *CVI_ISP_SetDRCAttr* : Set DRC attribute parameter.
- *CVI_ISP_GetDRCAttr* : Get DRC attribute parameter.

31.2.1 CVI_ISP_SetDRCAttr

【Description】

Set DRC attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetDRCAttr(VI_PIPE ViPipe, const ISP_DRC_ATTR_S *pstDRCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDRCAttr	DRC attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- When the wdr mode is enabled, you are advised to set Gamma table to sRGB.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDRCAttr*

31.2.2 CVI_ISP_GetDRCAttr

【Description】

Get DRC attribute parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetDRCAttr(VI_PIPE ViPipe, ISP_DRC_ATTR_S *pstDRCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDRCAttr	DRC attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

- When the wdr mode is enabled, you are advised to set Gamma table to sRGB..

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDRCAttr*

31.3 Data Types

- *ISP_DRC_MANUAL_ATTR_S* : DRC attribute manual parameter.
- *ISP_DRC_AUTO_ATTR_S* : DRC attribute auto parameter.
- *ISP_DRC_ATTR_S* : DRC attribute parameter.

31.3.1 ISP_DRC_MANUAL_ATTR_S

【Description】

DRC attribute manual parameter.

【Syntax】

```
typedef struct _ISP_DRC_MANUAL_ATTR_S {
 CVI_U32 TargetYScale; /*RW; Range:[0x0, 0x800]*/
 CVI_U16 HdrStrength; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 DEAdaptPercentile; /*RW; Range:[0x0, 0x19]*/
 CVI_U8 DEAdaptTargetGain; /*RW; Range:[0x1, 0x60]*/
 CVI_U8 DEAdaptGainUB; /*RW; Range:[0x1, 0xff]*/
 CVI_U8 DEAdaptGainLB; /*RW; Range:[0x1, 0xff]*/
 CVI_U8 BritInflectPtLuma; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 BritContrastLow; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 BritContrastHigh; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 SdrTargetY; /*RW; Range:[0x0, 0xff]*/
 CVI_U8 SdrTargetYGain; /*RW; Range:[0x20, 0x80]*/
 CVI_U16 SdrGlobalToneStr; /*RW; Range:[0x0, 0x100]*/
 CVI_U8 SdrDEAdaptPercentile; /*RW; Range:[0x0, 0x19]*/
 CVI_U8 SdrDEAdaptTargetGain; /*RW; Range:[0x1, 0x40]*/
 CVI_U8 SdrDEAdaptGainLB; /*RW; Range:[0x1, 0xff]*/
 CVI_U8 SdrDEAdaptGainUB; /*RW; Range:[0x1, 0xff]*/
 CVI_U8 SdrBritInflectPtLuma; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 SdrBritContrastLow; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 SdrBritContrastHigh; /*RW; Range:[0x0, 0x64]*/
 CVI_U8 TotalGain; /*RW; Range:[0x0, 0xff]*/
} ISP_DRC_MANUAL_ATTR_S;
```

【Members】

Member	Description
LETargetYScale	The tone map of the dark part enhances the reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32

continues on next page

Table 31.1 – continued from previous page

Member	Description
TargetYScale	Reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
HdrStrength	Control the parameter of the HDR enhancement effect. The larger the value, the stronger the overall transparency stretch, and vice versa, the weaker the stretch. Value range: [0x0, 0x100] Data type: CVI_U16
DarkOffset	Control the brightening of dark areas, the larger the value, the more brightening of dark areas, but the transparency of bright areas will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x100] Data type: CVI_U16
ContrastDarkMinThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastDarkMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMinThrd	Local contrast enhancement: The weight used to control the contrast of bright areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of bright areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16

continues on next page

Table 31.1 – continued from previous page

Member	Description
ContrastBrightMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastGain	Global contrast control parameter, 128 is 1 times, the smaller the value is less than 128, the more the contrast will be reduced, and the larger the value is greater than 128, the more the contrast will be improved. Value range: [0x0, 0x100] Data type: CVI_U16
DarkToneRange	Control the brightening range value of the dark area. The larger the value is, the brighter the dark area will be, but the transparency of the bright area will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x100] Data type: CVI_U16
DEAdaptPercentile	Defines the percentile for Dark, which determines the areas not to be enhanced. Value range: [0x0, 0x19] Data type: CVI_U8
DEAdaptTargetGain	The enhancement of Dark Tone adaptive target, the bigger the value is, the brighter it will be, 32 is double, 40 is 1.25x. Value range: [0x1, 0x40] Data type: CVI_U8
DEAdaptGainLB	The lower bound of Dark Tone adaptive enhancement, the larger the value, the less dark it will be suppressed, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
DEAdaptGainUB	The upper limit of Dark Tone adaptive enhancement, the larger the value, the brighter it is, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
SdrDEAdaptPercentile	Defines the percentile for Dark, which determines the areas not to be enhanced. Value range: [0x0, 0x19] Data type: CVI_U8

continues on next page

Table 31.1 – continued from previous page

Member	Description
SdrDEAdaptTargetGain	The enhancement of Dark Tone adaptive target, the bigger the value is, the brighter it will be, 32 is double, 40 is 1.25x. Value range: [0x1, 0x40] Data type: CVI_U8
SdrDEAdaptGainLB	The lower bound of Dark Tone adaptive enhancement, the larger the value, the less dark it will be suppressed, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
SdrDEAdaptGainUB	The upper limit of Dark Tone adaptive enhancement, the larger the value, the brighter it is, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
BritInflectPtLuma	Bright Tone The brightness of the long-short exposure boundary area, the larger the value, the higher the brightness. Value range: [0x0, 0x64] Data type: CVI_U8
BritContrastLow	Bright Tone The degree of darkening of dark areas in bright areas, the larger the value, the more darkening. Value range: [0x0, 0x64] Data type: CVI_U8
BritContrastHigh	Bright Tone Brightening degree of the bright areas in the bright area, the larger the value, the more the brightening. Value range: [0x0, 0x64] Data type: CVI_U8
SdrTargetY	Globally use Global Tone to brighten the screen, the larger the value, the more the brightness will be raised. Value range: [0x0, 0xFF] Data type: CVI_U8
SdrTargetYGain	Globally use Global Tone to brighten the screen, and the target brightness is a multiple of the current average brightness, 1x=32, 2x = 64. Value range: [0x20, 0x80] Data type: CVI_U8
SdrGlobalToneStr	The strength of the Global Tone, the larger the value, the stronger the Global Tone, otherwise it is closer to the Linear Tone. Value range: [0x0, 0x100] Data type: CVI_U16
SdrBritInflectPtLuma	Bright Tone The brightness of the long-short exposure boundary area, the larger the value, the higher the brightness. Value range: [0x0, 0x64] Data type: CVI_U8

continues on next page

Table 31.1 – continued from previous page

Member	Description
SdrBritContrastLow	Bright Tone The degree of darkening of dark areas in bright areas, the larger the value, the more darkening. Value range: [0x0, 0x64] Data type: CVI_U8
SdrBritContrastHigh	Bright Tone Brightening degree of the bright areas in the bright area, the larger the value, the more the brightening. Value range: [0x0, 0x64] Data type: CVI_U8
LETargetYScale	The tone map of the dark part enhances the reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
TargetYScale	Reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
HdrStrength	Control the parameter of the HDR enhancement effect. The larger the value, the stronger the overall transparency stretch, and vice versa, the weaker the stretch. Value range: [0x0, 0xff] Data type: CVI_U16
DarkOffset	Control the brightening of dark areas, the larger the value, the more brightening of dark areas, but the transparency of bright areas will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8

continues on next page

Table 31.1 – continued from previous page

Member	Description
ContrastDarkMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMinThrd	Local contrast enhancement: The weight used to control the contrast of bright areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of bright areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
TotalGain	Overall detail intensity, 1x = 32 Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDRCAttr*
- *CVI_ISP_GetDRCAttr*

31.3.2 ISP_DRC_AUTO_ATTR_S

【Description】

DRC attribute auto parameter.

【Syntax】

```
typedef struct _ISP_DRC_AUTO_ATTR_S {
 CVI_U32 TargetYScale[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x800]*/
```

(continues on next page)

(continued from previous page)

```
  CVI_U16 HdrStrength[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
  CVI_U8 DEAdaptPercentile[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x19]*/
  CVI_U8 DEAdaptTargetGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0x60]*/
  CVI_U8 DEAdaptGainUB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
  CVI_U8 DEAdaptGainLB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
  CVI_U8 BritInflectPtLuma[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 BritContrastLow[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 BritContrastHigh[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 SdrTargetY[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0xff]*/
  CVI_U8 SdrTargetYGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x20, 0x80]*/
  CVI_U16 SdrGlobalToneStr[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x100]*/
  CVI_U8 SdrDEAdaptPercentile[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x19]*/
  CVI_U8 SdrDEAdaptTargetGain[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0x40]*/
  CVI_U8 SdrDEAdaptGainLB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
  CVI_U8 SdrDEAdaptGainUB[ISP_AUTO_LV_NUM]; /*RW; Range:[0x1, 0xff]*/
  CVI_U8 SdrBritInflectPtLuma[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 SdrBritContrastLow[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 SdrBritContrastHigh[ISP_AUTO_LV_NUM]; /*RW; Range:[0x0, 0x64]*/
  CVI_U8 TotalGain[ISP_AUTO_ISO_STRENGTH_NUM]; /*RW; Range:[0x0, 0xff]*/
} ISP_DRC_AUTO_ATTR_S;
```

【Members】

Member	Description
LETargetYScale	The tone map of the dark part enhances the reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
TargetYScale	Reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
HdrStrength	Control the parameter of the HDR enhancement effect. The larger the value, the stronger the overall transparency stretch, and vice versa, the weaker the stretch. Value range: [0x0, 0x100] Data type: CVI_U16

continues on next page

Table 31.2 – continued from previous page

Member	Description
DarkOffset	Control the brightening of dark areas, the larger the value, the more brightening of dark areas, but the transparency of bright areas will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x100] Data type: CVI_U16
ContrastDarkMinThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastDarkMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMinThrd	Local contrast enhancement: The weight used to control the contrast of bright areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of bright areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8

continues on next page

Table 31.2 – continued from previous page

Member	Description
ContrastGain	Global contrast control parameter, 128 is 1 times, the smaller the value is less than 128, the more the contrast will be reduced, and the larger the value is greater than 128, the more the contrast will be improved. Value range: [0x0, 0x100] Data type: CVI_U16
DarkToneRange	Control the brightening range value of the dark area. The larger the value is, the brighter the dark area will be, but the transparency of the bright area will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x100] Data type: CVI_U16
DEAdaptPercentile	Defines the percentile for Dark, which determines the areas not to be enhanced. Value range: [0x0, 0x19] Data type: CVI_U8
DEAdaptTargetGain	The enhancement of Dark Tone adaptive target, the bigger the value is, the brighter it will be, 32 is double, 40 is 1.25x. Value range: [0x1, 0x40] Data type: CVI_U8
DEAdaptGainLB	The lower bound of Dark Tone adaptive enhancement, the larger the value, the less dark it will be suppressed, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
DEAdaptGainUB	The upper limit of Dark Tone adaptive enhancement, the larger the value, the brighter it is, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
SdrDEAdaptPercentile	Defines the percentile for Dark, which determines the areas not to be enhanced. Value range: [0x0, 0x19] Data type: CVI_U8
SdrDEAdaptTargetGain	The enhancement of Dark Tone adaptive target, the bigger the value is, the brighter it will be, 32 is double, 40 is 1.25x. Value range: [0x1, 0x40] Data type: CVI_U8
SdrDEAdaptGainLB	The lower bound of Dark Tone adaptive enhancement, the larger the value, the less dark it will be suppressed, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8

continues on next page

Table 31.2 – continued from previous page

Member	Description
SdrDEAdaptGainUB	The upper limit of Dark Tone adaptive enhancement, the larger the value, the brighter it is, 32 is double, and 96 is 3x. Value range: [0x1, 0xFF] Data type: CVI_U8
BritInflectPtLuma	Bright Tone The brightness of the long-short exposure boundary area, the larger the value, the higher the brightness. Value range: [0x0, 0x64] Data type: CVI_U8
BritContrastLow	Bright Tone The degree of darkening of dark areas in bright areas, the larger the value, the more darkening. Value range: [0x0, 0x64] Data type: CVI_U8
BritContrastHigh	Bright Tone Brightening degree of the bright areas in the bright area, the larger the value, the more the brightening. Value range: [0x0, 0x64] Data type: CVI_U8
SdrTargetY	Globally use Global Tone to brighten the screen, the larger the value, the more the brightness will be raised. Value range: [0x0, 0xFF] Data type: CVI_U8
SdrTargetYGain	Globally use Global Tone to brighten the screen, and the target brightness is a multiple of the current average brightness, 1x=32, 2x = 64. Value range: [0x20, 0x80] Data type: CVI_U8
SdrGlobalToneStr	The strength of the Global Tone, the larger the value, the stronger the Global Tone, otherwise it is closer to the Linear Tone. Value range: [0x0, 0x100] Data type: CVI_U16
SdrBritInflectPtLuma	Bright Tone The brightness of the long-short exposure boundary area, the larger the value, the higher the brightness. Value range: [0x0, 0x64] Data type: CVI_U8
SdrBritContrastLow	Bright Tone The degree of darkening of dark areas in bright areas, the larger the value, the more darkening. Value range: [0x0, 0x64] Data type: CVI_U8
SdrBritContrastHigh	Bright Tone Brightening degree of the bright areas in the bright area, the larger the value, the more the brightening. Value range: [0x0, 0x64] Data type: CVI_U8

continues on next page

Table 31.2 – continued from previous page

Member	Description
LETargetYScale	The tone map of the dark part enhances the reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
TargetYScale	Reference target brightness value gain, drc generates a reference target brightness value based on this value and the average value of the current wide dynamic brightness distribution, and adaptively generates a tone mapping curve based on this value Value range: [0x0, 0x800] Data type: CVI_U32
HdrStrength	Control the parameter of the HDR enhancement effect. The larger the value, the stronger the overall transparency stretch, and vice versa, the weaker the stretch. Value range: [0x0, 0xff] Data type: CVI_U16
DarkOffset	Control the brightening of dark areas, the larger the value, the more brightening of dark areas, but the transparency of bright areas will decrease. It is not recommended to set this value too large. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of dark areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastDarkMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastDarkMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of dark area contrast. Value range: [0x0, 0xff] Data type: CVI_U8

continues on next page

Table 31.2 – continued from previous page

Member	Description
ContrastBrightMinThrd	Local contrast enhancement: The weight used to control the contrast of bright areas adapts to the minimum threshold with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMaxThrd	Local Contrast Enhancement: The weight used to control the contrast of bright areas adapts to the maximum critical value with the dynamic range. Value range: [0x0, 0x3ff] Data type: CVI_U16
ContrastBrightMinWeight	Local Contrast Enhancement: Used to control the weight corresponding to the minimum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
ContrastBrightMaxWeight	Local Contrast Enhancement: Used to control the weight corresponding to the maximum critical value of bright area contrast. Value range: [0x0, 0xff] Data type: CVI_U8
TotalGain	Overall detail intensity, 1x = 32 Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDRCAttr*
- *CVI_ISP_GetDRCAttr*

31.3.3 ISP_DRC_ATTR_S

【Description】

DRC attribute parameter.

【Syntax】

```
typedef struct _ISP_DRC_ATTR_S {
 CVI_BOOL Enable; /*RW; Range:[0, 1]*/
 ISP_OP_TYPE_E enOpType;
 CVI_U8 UpdateInterval; /*RW; Range:[0x1, 0xFF]*/
 CVI_U8 TuningMode; /*RW; Range:[0x0, 0x7]*/
 CVI_BOOL LocalToneEn; /*RW; Range:[0x0, 0x1]*/
 CVI_BOOL LocalToneRefineEn; /*RW; Range:[0x0, 0x1]*/
```

(continues on next page)

(continued from previous page)

```
    CVI_U8 ToneCurveSelect; /*RW; Range:[0x0, 0x1]*/
    CVI_U16 CurveUserDefine[DRC_GLOBAL_USER_DEFINE_NUM]; /*RW; Range:[0x0,␣
↪0xffff]*/
    CVI_U16 DarkUserDefine[DRC_DARK_USER_DEFINE_NUM]; /*RW; Range:[0x0, 0xffff]*/
    CVI_U16 BrightUserDefine[DRC_BRIGHT_USER_DEFINE_NUM]; /*RW; Range:[0x0,␣
↪0xffff]*/
    CVI_U32 ToneCurveSmooth; /*RW; Range:[0x0, 0x1f4]*/
    CVI_U8 CoarseFltScale; /*RW; Range:[0x3, 0x6]*/
    CVI_U8 SdrTargetYGainMode; /*RW; Range:[0x0, 0x1]*/
    CVI_BOOL DetailEnhanceEn; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 LumaGain[33]; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 DetailEnhanceMtIn[4]; /*RW; Range:[0x0, 0xff]*/
    CVI_U16 DetailEnhanceMtOut[4]; /*RW; Range:[0x0, 0x100]*/
    CVI_U8 OverShootThd; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThd; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 OverShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 UnderShootGain; /*RW; Range:[0x0, 0x3f]*/
    CVI_U8 OverShootThdMax; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 UnderShootThdMin; /*RW; Range:[0x0, 0xff]*/
    CVI_BOOL SoftClampEnable; /*RW; Range:[0x0, 0x1]*/
    CVI_U8 SoftClampUB; /*RW; Range:[0x0, 0xff]*/
    CVI_U8 SoftClampLB; /*RW; Range:[0x0, 0xff]*/
    CVI_BOOL dbg_182x_sim_enable; /*RW; Range:[0, 1]*/
    CVI_U8 DarkMapStr; /*RW; Range:[0x0, 0x80]*/
    CVI_U8 BritMapStr; /*RW; Range:[0x0, 0x80]*/
    CVI_U8 SdrDarkMapStr; /*RW; Range:[0x0, 0x80]*/
    CVI_U8 SdrBritMapStr; /*RW; Range:[0x0, 0x80]*/
    CVI_U32 DRCMu[32]; /*RW; Range:[0x0, 0x7fffffff]*/
    ISP_DRC_MANUAL_ATTR_S stManual;
    ISP_DRC_AUTO_ATTR_S stAuto;
} ISP_DRC_ATTR_S;
```

【Members】

Member	Description
Enable	DRC module enable Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8

continues on next page

Table 31.3 – continued from previous page

Member	Description
TuningMode	Debug mode, output visual auxiliary information to help users debug. 0: Do not output visual auxiliary information 1: Output the visual result of SE image highlight information. 2: Output the visualization result of the dark part information of the LE screen. 3: Output the SE screen DC information visualization result. 4: Output LE screen DC information visualization results Value range: [0x0, 0x7] Data type: CVI_U8
LocalToneEn	Enable Bright Local Tone Mapping and Dark Local Tone Mapping Value range: [0x0, 0x1] Data type: CVI_BOOL
LocalToneRefineEn	Optimize the enablement of local tone mapping to make the area division more precise Value range: [0x0, 0x1] Data type: CVI_BOOL
CoarseFltScale	Control the size of the filtering window, and the adjustment step size is larger than that of DRangeFltScale. The larger the value, the larger the filter window, the richer the details. On the contrary, the weaker Value range: [0x3, 0x6] Data type: CVI_U8
DarkUserDefine[257]	User-Definable Shadow Tone Mapping Curves Value range: [0x0, 0xffff] Data type: CVI_U16
BrightUserDefine[513]	User-Definable Highlight Tone Mapping Curve Value range: [0x0, 0xffff] Data type: CVI_U16
CurveUserDefine[1025]	user defined curve Value range: [0x0, 0xffff] Data type: CVI_U16
ToneCurveSelect	Tone Curve selection: 0: Select user-defined curve. 1: Select the Asymmetry curve Value range: [0x0, 0x1] Data type: CVI_U8
LinearStart	Limits the intensity of tone-mapping compression within a specified value range, preserving the transparency of image values falling within that region. LinearStart defines the starting point value of the range Value range: [0x0, 0xfff] Data type: CVI_U16

continues on next page

Table 31.3 – continued from previous page

Member	Description
LinearEnd	Limits the intensity of tone-mapping compression within a specified value range, preserving the transparency of image values falling within that region. LinearEnd defines the end value of the range Value range: [0x0, 0xfff] Data type: CVI_U16
ToneCurveSmooth	The smoothness of the change in the time domain of the Tone Curve curve. The larger the value, the smoother the change in the time domain, and vice versa Value range: [0x0, 0x1f4] Data type: CVI_U32
DetailEnhanceEnable	Sharpen module enable Value range: [0x0, 0x1] Data type: CVI_BOOL
DetailEnhanceEn	Enable the ltm_ee module to enhance the details of HDR Value range: [0x0, 0x1] Data type: CVI_BOOL
LumaGain[33]	Adjust the intensity of enhanced details according to the brightness, 1x = 64 Value range: [0x0, 0xff] Data type: CVI_U8
DetailEnhanceMtIn[4]	According to the strength of the motion, adjust the enhanced details Value range: [0x0, 0xff] Data type: CVI_U8
DetailEnhanceMtOut[4]	According to the strength of the motion, adjust the enhanced details Value range: [0x0, 0x100] Data type: CVI_U16
OverShootThd	White edge sharpening upper limit Value range: [0x0, 0xff] Data type: CVI_U8
UnderShootThd	Black edge sharpening upper limit range Value range: [0x0, 0xff] Data type: CVI_U8
OverShootGain	The intensity of white edge sharpening, 16 is doubled Value range: [0x0, 0x3f] Data type: CVI_U8
UnderShootGain	The intensity of black edge sharpening, 16 is doubled Value range: [0x0, 0x3f] Data type: CVI_U8
OverShootThdMax	White edge sharpening maximum upper limit range Value range: [0x0, 0xff] Data type: CVI_U8
UnderShootThdMin	Black edge sharpening maximum upper limit range Value range: [0x0, 0xff] Data type: CVI_U8

continues on next page

Table 31.3 – continued from previous page

Member	Description
SoftClampEnable	Smooth Processing Edge Enhancement Value range: [0x0, 0x1] Data type: CVI_BOOL
SoftClampUB	The upper and lower bounds of edge enhancement are smoothed. The larger the value is, the more continuous the edge enhancement will be, but the weaker the enhancement effect will be. Value range: [0x0, 0xff] Data type: CVI_U8
SoftClampLB	The upper and lower bounds of edge enhancement are smoothed. The larger the value is, the more continuous the edge enhancement will be, but the weaker the enhancement effect will be. Value range: [0x0, 0xff] Data type: CVI_U8
dbg_182x_sim_enable	Debug mode for 182x performance simulation Value range: [0x0, 0x1] Data type: CVI_BOOL
DrcMode	DRC Mode 0: Do DRC in HSV domain 1: Do DRC in RGB domain Value range: [0x0, 0x1] Data type: CVI_BOOL
SatEnable	Saturation debug function enable Value range: [0x0, 0x1] Data type: CVI_BOOL
SdrToneCurveSelect	SDR DRC Tone curve curve selection 0: Bypass Mode 4: SdrDarkEnhance(SdrDE) Mode Value range: [0x0, 0x4] Data type: CVI_U8
DEAdaptEn	Dark Tone Adaptive Function Enabled Value range: [0, 1] Data type: CVI_BOOL
DEAdaptMode	Dark Tone adaptive function mode 0: Do not consider Dark info 1: Consider Dark info Value range: [0, 1] Data type: CVI_BOOL
SdrDEAdaptMode	Dark Tone adaptive function mode 0: Do not consider Dark info 1: Consider Dark info Value range: [0, 1] Data type: CVI_BOOL

continues on next page

Table 31.3 – continued from previous page

Member	Description
DarkMapStr	The strength of Dark Tone, it is recommended to use the default value, the higher the value, the stronger the effect of dark tone Value range: [0x1, 0x80] Data type: CVI_U8
SdrDarkMapStr	The strength of Dark Tone, it is recommended to use the default value, the higher the value, the stronger the effect of dark tone Value range: [0x1, 0x80] Data type: CVI_U8
BritMapStr	The strength of Bright Tone, it is recommended to use the default value, the higher the value, the stronger the effect of bright tone Value range: [0x1, 0x80] Data type: CVI_U8
SdrBritMapStr	The strength of Bright Tone, it is recommended to use the default value, the higher the value, the stronger the effect of bright tone Value range: [0x1, 0x80] Data type: CVI_U8
SdrTargetYGainMode	SdrTargetY Gain Mode switch 0: Directly specify the average target brightness of the screen LmapLeMode 1: Based on the average of the screen, the brightening factor is 1x=32, 2x=64 Value range: [0x0, 0x1] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDRCAttr*
- *CVI_ISP_GetDRCAttr*

32 DIS

32.1 Function Overview

DIS module calculates the current displacement by comparing the current image and the previous frame image, and it is transferred to the post module for image cutting, and then the anti-jitter effect is realized.

32.2 API Reference

- *CVI_ISP_SetDISAttr* : Set DIS property parameter.
- *CVI_ISP_GetDISAttr* : Get DIS property parameter.
- *CVI_ISP_SetDISConfig* : Set DIS control parameter.
- *CVI_ISP_GetDISConfig* : Get DIS control parameter.

32.2.1 CVI_ISP_SetDISAttr

【Description】

Set DIS property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetDisAttr(VI_PIPE ViPipe, const ISP_DIS_ATTR_S *pstDisAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDisAttr	DIS property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDISAttr*

32.2.2 CVI_ISP_GetDISAttr

【Description】

Get DIS property parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetDisAttr(VI_PIPE ViPipe, ISP_DIS_ATTR_S *pstDisAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDisAttr	DIS property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDISAttr*

32.2.3 CVI_ISP_SetDISConfig

【Description】

Set DIS control parameter.

【Syntax】

```
CVI_S32 CVI_ISP_SetDisConfig(VI_PIPE ViPipe, const ISP_DIS_CONFIG_S *pstDisConfig);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDisConfig	DIS control parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetDISConfig*

32.2.4 CVI_ISP_GetDISConfig

【Description】

Get DIS control parameter.

【Syntax】

```
CVI_S32 CVI_ISP_GetDisConfig(VI_PIPE ViPipe, const ISP_DIS_CONFIG_S␣
↪*pstDisConfig);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstDisConfig	DIS control parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetDISConfig*

32.3 Data Types

- *ISP_DIS_ATTR_S* : DIS attribute parameter.
- *ISP_DIS_CONFIG_S* : DIS configuration parameter.
- *DIS_MODE_E* : DIS mode parameter
- *DIS_MOTION_LEVEL_E* : DIS level parameter

32.3.1 ISP_DIS_ATTR_S

【Description】

DIS attribute parameter.

【Syntax】

```
typedef struct _ISP_DIS_ATTR_S {
   CVI_BOOL enable;
   CVI_U32 movingSubjectLevel;
   CVI_U32 horizontalLimit;
   CVI_U32 verticalLimit;
   CVI_BOOL stillCrop;
} ISP_DIS_ATTR_S;
```

【Member】

Member	Description
enable	DIS enable switch; CVI_FALSE: not enabled CVI_TRUE: Enable
movingSubjectLevel	A parameter used to determine whether an object is moving or not; the smaller the value of the parameter used to judge whether an object is moving or not, the better the anti-shake effect is, but it is prone to offset. On the contrary, the jitter is easy to increase, but the offset phenomenon is reduced. Value range: [0, 6] Data type: CVI_U32
horizontalLimit	Horizontal displacement limit; when the displacement of a large area object causes the background displacement to reach a certain range, the anti- shake effect is not done. The calculation method is image_width * horizontalLimit / 1000 Value range: [0, 1000] Data type: CVI_U32
verticalLimit	Vertical displacement limit; when the displacement of a large area object causes the background displacement to reach a certain range, the anti-shake effect is not done. The calculation method is image_height * verticalLimit / 1000 Value range: [0, 1000] Data type: CVI_U32
stillCrop	Turn off the anti-shake effect of DIS, but keep the output of clipping ratio to avoid the change of image size CVI_FALSE: not enabled CVI_TRUE: Enable

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDisAttr*
- *CVI_ISP_GetDisAttr*

32.3.2 ISP_DIS_CONFIG_S

【Description】

DIS configuration parameter.

【Syntax】

```
typedef struct _ISP_DIS_CONFIG_S {
DIS_MODE_E enMode;
DIS_MOTION_LEVEL_E enMotionLevel;
CVI_U32 u32CropRatio;
} ISP_DIS_CONFIG_S;
```

【Member】

Member	Description
mode	Digital anti jitter algorithm mode; only one algorithm mode is supported at present Value range: [0, 0] Data type: dis_MODE_E
motionLevel	The motion level of camera only supports IS_MOTION_LEVEL_NORMAL at present. Value range: [1,1] Data type: DIS_MOTION_LEVEL
cropRatio	Image clipping ratio of DIS output; the larger the cut ratio of DIS output image, the larger the retained image, and vice versa. The recommended default value is 94 Value range: [50, 98] Data type: CVI_U32

【Note】

CropRatio will affect the maximum displacement that DIS algorithm can support. Assuming that CropRatio is 98, it means that only 2% of the wide energy is left for displacement.

【Related Data Type and Interface】

- *CVI_ISP_SetDisConfig*
- *CVI_ISP_GetDisConfig*

32.3.3 DIS_MODE_E

【Description】

Define DIS anti-shake algorithm mode.

【Syntax】

```
typedef enum _ISP_DIS_MODE_E {
   DIS_MODE_2_DOF_GME = 0, /* Only use with GME in 2 dof */
   DIS_MODE_DEBUG,
   DIS_MODE_DOF_BUTT,
} DIS_MODE_E;
```

【Member】

Member	Description
DIS_MODE_2_DOF_GME	Two degree of freedom GME algorithm
DIS_MODE_DEBUG	debug mode

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDisConfig*
- *CVI_ISP_GetDisConfig*

32.3.4 DIS_MOTION_LEVEL_E

【Description】

Define the motion level of the lens.

【Syntax】

```
typedef enum _ISP_DIS_MOTION_LEVEL_E {
   DIS_MOTION_LEVEL_NORMAL = 1,
   DIS_MOTION_LEVEL_BUTT
} DIS_MOTION_LEVEL_E;
```

【Member】

Member	Description
DIS_MOTION_LEVEL_NORMAL	Normal level of lens wobble

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetDisConfig*
- *CVI_ISP_GetDisConfig*

33 Mono

33.1 Function Overview

This section describes how to set monochrome modes and properties.

33.2 API Reference

- *CVI_ISP_SetMonoAttr* : Set Mono *attribute* parameter
- *CVI_ISP_GetMonoAttr* : Get Mono *attribute* parameter

33.2.1 CVI_ISP_SetMonoAttr

【Description】

Set Mono attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetMonoAttr(VI_PIPE ViPipe, const ISP_MONO_ATTR_S *pstMonoAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstMonoAttr	*Mono attribute parameter*	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetMonoAttr*

33.2.2 CVI_ISP_GetMonoAttr

【Description】

Get Mono attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetMonoAttr(VI_PIPE ViPipe, ISP_MONO_ATTR_S *pstMonoAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstMonoAttr	Mono attribute parameter	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetMonoAttr*

33.3 Data Types

- *ISP_MONO_ATTR_S* : Mono attribute parameter

33.3.1 ISP_MONO_ATTR_S

【Description】

Mono attribute parameter

【Syntax】

```
typedef struct _ISP_MONO_ATTR_S {
    CVI_BOOL Enable;
    CVI_U8 UpdateInterval;
} ISP_MONO_ATTR_S;
```

【Member】

Member	Description
Enable	Monochrome mode enabled Value range: [0x0, 0x1] Data type: CVI_BOOL
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetMonoAttr*
- *CVI_ISP_GetMonoAttr*

34 YCONTRAST

34.1 Function Overview

Y domain linear contrast.

34.2 API Reference

- *CVI_ISP_SetYContrastAttr* : Set Y domain contrast attribute parameter
- *CVI_ISP_GetYContrastAttr* : Get Y domain contrast attribute parameter

34.2.1 CVI_ISP_SetYContrastAttr

【Description】

Set Y range contrast attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetYContrastAttr(VI_PIPE ViPipe, const ISP_YCONTRAST_ATTR_S
↪*pstYContrastAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYContrastAttr	Y domain contrast attribute parameter	Input

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetYContrastAttr*

34.2.2 CVI_ISP_GetYContrastAttr

【Description】

Get Y range contrast attribute parameter

【Syntax】

```
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstYContrastAttr	Y domain contrast attribute parameter	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetYContrastAttr*

34.3 Data Types

- *ISP_YCONTRAST_MANUAL_ATTR_S* : Gamma attribute manual parameter
- *ISP_YCONTRAST_AUTO_ATTR_S* : gamma attribute automatic parameter
- *ISP_YCONTRAST_ATTR_S* : gamma attribute parameter

34.3.1 ISP_YCONTRAST_MANUAL_ATTR_S

【Description】

Gamma attribute manual parameter

【Syntax】

```
typedef struct _ISP_YCONTRAST_MANUAL_ATTR_S {
 CVI_U8 ContrastLow;
 CVI_U8 ContrastHigh;
 CVI_U8 CenterLuma;
} ISP_YCONTRAST_MANUAL_ATTR_S;
```

【Member】

Member	Description
ContrastLow	Contrast strength of areas smaller than the center point. The larger the value, the stronger the contrast. Value range: [0x0, 0x64] Data type: CVI_U8
ContrastHigh	The contrast strength of the area greater than the center point position. The larger the value, the stronger the contrast. Value range: [0x0, 0x64] Data type: CVI_U8
CenterLuma	The position of the center point will strengthen the contrast from the center point to both sides. Value range: [0x0, 0x40] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYContrastAttr*
- *CVI_ISP_GetYContrastAttr*

34.3.2 ISP_YCONTRAST_AUTO_ATTR_S

【Description】

gamma attribute automatic parameter

【Syntax】

```
typedef struct _ISP_YCONTRAST_AUTO_ATTR_S {
  CVI_U8 ContrastLow[ISP_AUTO_LV_NUM];
  CVI_U8 ContrastHigh[ISP_AUTO_LV_NUM];
  CVI_U8 CenterLuma[ISP_AUTO_LV_NUM];
} ISP_YCONTRAST_AUTO_ATTR_S;
```

【Member】

Member	Description
ContrastLow	Contrast strength of areas smaller than the center point. The larger the value, the stronger the contrast. Value range: [0x0, 0x64] Data type: CVI_U8
ContrastHigh	The contrast strength of the area greater than the center point position. The larger the value, the stronger the contrast. Value range: [0x0, 0x64] Data type: CVI_U8
CenterLuma	The position of the center point will strengthen the contrast from the center point to both sides. Value range: [0x0, 0x40] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYContrastAttr*
- *CVI_ISP_GetYContrastAttr*

34.3.3 ISP_YCONTRAST_ATTR_S

【Description】

gamma attribute parameter

【Syntax】

```
typedef struct _ISP_YCONTRAST_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
```

(continues on next page)

(continued from previous page)

```
  ISP_YCONTRAST_MANUAL_ATTR_S stManual;
  ISP_YCONTRAST_AUTO_ATTR_S stAuto;
} ISP_YCONTRAST_ATTR_S;
```

【Member】

Member	Description
Enable	The YCONTRAST module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data type: CVI_BOOL
enOpType	job type OP_TYPE_AUTO: automatic mode OP_TYPE_MANUAL: manual mode
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetYContrastAttr*
- *CVI_ISP_GetYContrastAttr*

35 CA

35.1 Function Overview

Adjust UV domain saturation. This module is divided into CA mode and CP (thermal imaging) mode. Only one can be opened at the same time.

CA mode: The saturation is adjusted based on input Y, so it can be adjusted locally.

CP mode: Because the thermal imaging mode only has Y value, it will search the preset debug color template according to the Y value, search the corresponding YUV value, and make it color.

35.2 API Reference

- *CVI_ISP_SetCAAttr* : set saturation attribute parameter
- *CVI_ISP_GetCAAttr* : Get the saturation attribute parameter

35.2.1 CVI_ISP_SetCAAttr

【Description】

Set saturation attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetCAAttr(VI_PIPE ViPipe, const ISP_CA_ATTR_S *pstCAAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCAAttr	Saturation property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCAAttr*

35.2.2 CVI_ISP_GetCAAttr

【Description】

Get the saturation attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetCAAttr(VI_PIPE ViPipe, ISP_CA_ATTR_S *pstCAAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCAAttr	Saturation property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCAAttr*

35.3 Data Types

- *ISP_CA_MANUAL_ATTR_S* : Saturation attribute manual parameter
- *ISP_CA_AUTO_ATTR_S* : Saturation attribute automatic parameter
- *ISP_CA_ATTR_S* : Saturation attribute parameter

35.3.1 ISP_CA_MANUAL_ATTR_S

【Description】

Saturation attribute manual parameter

【Syntax】

```
typedef struct _ISP_CA_MANUAL_ATTR_S {
  CVI_U16 ISORatio;
  CVI_U16 YRatioLut[CA_LUT_NUM];
} ISP_CA_MANUAL_ATTR_S;
```

【Member】

Member	Description
ISORatio	CA mode, find UV gain based on ISO value. The UV adjustment gain of all pixels is the same. It is recommended that the gain be set larger at low ISO, and smaller at high ISO to suppress color noise in dark areas. Value range: [0x0, 0x7ff] Data Type: CVI_U16
YRatioLut[CA_LUT_NUM]	In CA mode, look up UV gain based on brightness Y. This value can set different UV gains according to different brightness levels. It is recommended that the gain in the bright area be set higher, and the color will be more vivid, while the gain in the dark area can be set smaller to suppress the color noise in the dark area. Value range: [0x0, 0x7ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCAAttr*
- *CVI_ISP_GetCAAttr*

35.3.2 ISP_CA_AUTO_ATTR_S

【Description】

Saturation attribute automatic parameter

【Syntax】

```
typedef struct _ISP_CA2_AUTO_ATTR_S {
  CVI_U16 ISORatio[ISP_AUTO_ISO_STRENGTH_NUM];
  CVI_U16 YRatioLut[CA_LUT_NUM][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CA2_AUTO_ATTR_S;
```

【Member】

Member	Description
ISORatio	CA mode, find UV gain based on ISO value. The UV adjustment gain of all pixels is the same. It is recommended that the gain be set larger at low ISO, and smaller at high ISO to suppress color noise in dark areas. Value range: [0x0, 0x7ff] Data Type: CVI_U16
YRatioLut[CA_LUT_NUM]	In CA mode, look up UV gain based on brightness Y. This value can set different UV gains according to different brightness levels. It is recommended that the gain in the bright area be set higher, and the color will be more vivid, while the gain in the dark area can be set smaller to suppress the color noise in the dark area. Value range: [0x0, 0x7ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCAAttr*
- *CVI_ISP_GetCAAttr*

35.3.3 ISP_CA_ATTR_S

【Description】

Saturation attribute parameter

【Syntax】

```
typedef struct _ISP_CA_ATTR_S {
  CVI_BOOL Enable;
  ISP_OP_TYPE_E enOpType;
  CVI_U8 UpdateInterval;
  CVI_BOOL CaCpMode;
  CVI_U8 CPLutY[CA_LUT_NUM];
  CVI_U8 CPLutU[CA_LUT_NUM];
  CVI_U8 CPLutV[CA_LUT_NUM];
  ISP_CA_MANUAL_ATTR_S stManual;
  ISP_CA_AUTO_ATTR_S stAuto;
} ISP_CA_ATTR_S;
```

【Member】

Member	Description
Enable	The CA module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data type: CVI_U8
CaCpMode	CA mode selection: 0: CA mode. 1: CP mode. Value range: [0, 1] Data Type: CVI_BOOL
CPLutY	In CP mode, look up the Y value corresponding to the LUT according to the brightness Y. Value range: [0, 0xff] Data type: CVI_U8
CPLutU	In CP mode, look up the U value corresponding to the LUT according to the brightness Y. Value range: [0, 0xff] Data type: CVI_U8
CPLutV	In CP mode, look up the V value corresponding to the LUT according to the brightness Y. Value range: [0, 0xff] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- CVI_ISP_SetCA2Attr
- CVI_ISP_GetCA2Attr

36 CA2

36.1 Function Overview

Adjust the UV domain saturation, readjust the saturation level based on the saturation of the input pixel

36.2 API Reference

- *CVI_ISP_SetCA2Attr* : Set the saturation attribute parameter
- *CVI_ISP_GetCA2Attr* : Get the saturation attribute parameter

36.2.1 CVI_ISP_SetCA2Attr

【Description】

Set the saturation attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetCA2Attr(VI_PIPE ViPipe, const ISP_CA2_ATTR_S *pstCA2Attr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCA2Attr	Saturation property parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCA2Attr*

36.2.2 CVI_ISP_GetCA2Attr

【Description】

Get the saturation attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetCA2Attr(VI_PIPE ViPipe, ISP_CA2_ATTR_S *pstCA2Attr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCA2Attr	Saturation property parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCA2Attr*

36.3 Data Types

- *ISP_CA2_MANUAL_ATTR_S* : Saturation attribute manual parameter
- *ISP_CA2_AUTO_ATTR_S* : Saturation attribute automatic parameter
- *ISP_CA2_ATTR_S* : Saturation attribute parameter

36.3.1 ISP_CA2_MANUAL_ATTR_S

【Description】

Saturation attribute manual parameter

【Syntax】

```
typedef struct _ISP_CA2_MANUAL_ATTR_S {
   CVI_U16 Ca2In[CA_LITE_NODE];
   CVI_U16 Ca2Out[CA_LITE_NODE];
} ISP_CA2_MANUAL_ATTR_S;
```

【Member】

Member	Description
Ca2In[CA_LITE_NODE]	An array of six values that determine the input saturation level Value range: [0x0, 0xc0] Data type: CVI_U8
Ca2Out[CA_LITE_NODE]	An array of six values defining the UV gain for the output. Find the UV gain according to the input saturation, the larger the value, the higher the saturation; otherwise, the smaller Value range: [0x0, 0x7ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCA2Attr*
- *CVI_ISP_GetCA2Attr*

36.3.2 ISP_CA2_AUTO_ATTR_S

【Description】

Saturation attribute automatic parameter

【Syntax】

```
typedef struct _ISP_CA2_AUTO_ATTR_S {
   CVI_U16 Ca2In[CA_LITE_NODE][ISP_AUTO_ISO_STRENGTH_NUM];
   CVI_U16 Ca2Out[CA_LITE_NODE][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CA2_AUTO_ATTR_S;
```

【Member】

Member	Description
Ca2In[CA_LITE_NODE]	Ca2In[CA_LITE_NODE] An array of six values that determine the input saturation level Value range: [0x0, 0xc0] Data type: CVI_U8
Ca2Out[CA_LITE_NODE]	Ca2Out[CA_LITE_NODE] An array of six values defining the output UV gain. Find the UV gain according to the input saturation, the larger the value, the higher the saturation; otherwise, the smaller Value range: [0x0, 0x7ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCA2Attr*
- *CVI_ISP_GetCA2Attr*

36.3.3 ISP_CA2_ATTR_S

【Description】

Saturation attribute parameter

【Syntax】

```
typedef struct _ISP_CA2_ATTR_S {
   CVI_BOOL Enable;
   ISP_OP_TYPE_E enOpType;
   CVI_U8 UpdateInterval;
   ISP_CA2_MANUAL_ATTR_S stManual;
```

(continues on next page)

(continued from previous page)

```
    ISP_CA2_AUTO_ATTR_S stAuto;
} ISP_CA2_ATTR_S;
```

【Member】

Member	Description
Enable	The CA2 module is enabled. Value range: [0, 1] Data Type: CVI_BOOL
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
stManual	Manual Mode Parameter Properties
stAuto	Auto Mode Parameter Properties

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCA2Attr*
- *CVI_ISP_GetCA2Attr*

37 CLUT

37.1 Function Overview

Linear conversion is done on RGB domain through a 17x17x17 3D LUT. The RGB pixel values can be interpolated into the table to obtain new RGB pixel values, which can be used to debug the color and brightness.

37.2 API Reference

- *CVI_ISP_SetClutAttr* : Set CLUT attribute parameter
- *CVI_ISP_GetClutAttr* : Get CLUT attribute parameter
- *CVI_ISP_SetClutSaturationAttr* : Set CLUT SbyS attribute parameter
- *CVI_ISP_GetClutSaturationAttr* : Get CLUT SbyS attribute parameter

37.2.1 CVI_ISP_SetClutAttr

【Description】

Set CLUT attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetClutAttr(VI_PIPE ViPipe, const ISP_CLUT_ATTR_S *pstClutAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCLUTAttr	CLUT attribute parameter	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files:cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【example】

None.

【Related Topic】

- *CVI_ISP_GetClutAttr*

37.2.2 CVI_ISP_GetClutAttr

【Description】

Get CLUT attribute parameter

【Syntax】

```
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCLUTAttr	CLUT attribute parameter	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files:cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【example】

None.

【Related Topic】

- *CVI_ISP_SetClutAttr*

37.2.3 CVI_ISP_SetClutSaturationAttr

【Description】

Set CLUT SbyS attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetClutSaturationAttr(VI_PIPE ViPipe, ISP_CLUT_SATURATION_ATTR_
↪S *pstClutSaturationAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstClutSaturationAttr	CLUT SbyS attribute parameter	Input

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files:cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【example】

None.

【Related Topic】

- *CVI_ISP_GetClutSaturationAttr*

37.2.4 CVI_ISP_GetClutSaturationAttr

【Description】

Get CLUT SbyS attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetClutSaturationAttr(VI_PIPE ViPipe, ISP_CLUT_SATURATION_ATTR_
↪S *pstClutSaturationAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstClutSaturationAttr	CLUT SbyS attribute parameter	Output

【Return Value】

Return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files:cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【example】

None.

【Related Topic】

- *CVI_ISP_SetClutSaturationAttr*

37.3 Data Types

- *ISP_CLUT_ATTR_S* : CLUT attribute parameter
- *ISP_CLUT_SATURATION_MANUAL_ATTR_S* : CLUT SbyS attribute manual parameter
- *ISP_CLUT_SATURATION_AUTO_ATTR_S* : CLUT SbyS attribute automatic parameter
- *ISP_CLUT_SATURATION_ATTR_S* : CLUT SbyS attribute parameter

37.3.1 ISP_CLUT_ATTR_S

【Description】

CLUT attribute parameter

【Syntax】

```
typedef struct _ISP_CLUT_ATTR_S {
   CVI_BOOL Enable;
   CVI_U8 UpdateInterval;
   CVI_U16 ClutR[ISP_CLUT_LUT_LENGTH];
   CVI_U16 ClutG[ISP_CLUT_LUT_LENGTH];
   CVI_U16 ClutB[ISP_CLUT_LUT_LENGTH];
} ISP_CLUT_ATTR_S;
```

【Member】

Member	Description
Enable	The CLUT module is enabled. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance. Value range: [0x0, 0xff] Data Type: CVI_U8
ClutR	LUT for the R channel. Value range: [0x0, 0x3ff] Data Type: CVI_U16
ClutG	LUT for the G channel. Value range: [0x0, 0x3ff] Data Type: CVI_U16
ClutB	LUT for the B channel. Value range: [0x0, 0x3ff] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetClutAttr*
- *CVI_ISP_GetClutAttr*

37.3.2 ISP_CLUT_SATURATION_MANUAL_ATTR_S

【Description】

CLUT SbyS attribute manual parameter

【Syntax】

```
typedef struct _ISP_CLUT_SATURATION_MANUAL_ATTR_S {
    CVI_U16 SatIn[4];
    CVI_U16 SatOut[4];
} ISP_CLUT_SATURATION_MANUAL_ATTR_S;
```

【Member】

Member	Description
SatIn [4]	SatIn [4] SbyS input Value range: [0, 0x2000] Data Type: CVI_U16
SatOut [4]	SatOut [4] SbyS output Value range: [0, 0x2000] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetClutSaturationAttr*
- *CVI_ISP_GetClutSaturationAttr*

37.3.3 ISP_CLUT_SATURATION_AUTO_ATTR_S

【Description】

CLUT SbyS attribute automatic parameter

【Syntax】

```
typedef struct _ISP_CLUT_SATURATION_AUTO_ATTR_S {
    CVI_U16 SatIn[4][ISP_AUTO_ISO_STRENGTH_NUM];
    CVI_U16 SatOut[4][ISP_AUTO_ISO_STRENGTH_NUM];
} ISP_CLUT_SATURATION_AUTO_ATTR_S;
```

【Member】

Member	Description
SatIn [4]	SatIn [4] SbyS input Value range: [0, 0x2000] Data Type: CVI_U16
SatOut [4]	SatOut [4] SbyS output Value range: [0, 0x2000] Data Type: CVI_U16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetClutSaturationAttr*
- *CVI_ISP_GetClutSaturationAttr*

37.3.4 ISP_CLUT_SATURATION_ATTR_S

【Description】

CLUT SbyS attribute parameter

【Syntax】

```
typedef struct _ISP_CLUT_SATURATION_ATTR_S {
   CVI_BOOL Enable;
   ISP_OP_TYPE_E enOpType;
   ISP_CLUT_SATURATION_MANUAL_ATTR_S stManual;
   ISP_CLUT_SATURATION_AUTO_ATTR_S stAuto;
} ISP_CLUT_SATURATION_ATTR_S;
```

【Member】

Member	Description
Enable	SbyS module enable. 0: off. 1: Enabled. Value range: [0, 1] Data Type: CVI_BOOL
enOpType	Select manual or automatic mode
stManual	Manual mode parameter attribute
stAuto	Auto mode parameter attribute

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetClutSaturationAttr*
- *CVI_ISP_GetClutSaturationAttr*

38 CSC

38.1 Function Overview

This feature provides a color gamut conversion of some related quality setting, by setting hue, luma, saturation, contrast, color gamut conversion standard specification types, affecting the effect of the picture, without needing to know what a transformation matrix calculation process, Of course, the user can also directly set the coeff and offset of the transformation matrix to affect the image quality

38.2 API Reference

- *CVI_ISP_SetCSCAttr* : Set CSC attribute parameter
- *CVI_ISP_GetCSCAttr* : Get CSC attribute parameter

38.2.1 CVI_ISP_SetCSCAttr

【Description】

Set CSC attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetCSCAttr(VI_PIPE ViPipe, const ISP_CSC_ATTR_S * pstCSCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCSCAttr	CSC attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetCSCAttr*

38.2.2 CVI_ISP_GetCSCAttr

【Description】

Get CSC attribute parameter

【Syntax】

```
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstCSCAttr	CSC attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetCSCAttr*

38.3 Data Types

- *ISP_CSC_ATTR_S* : CSC attribute parameter
- *ISP_CSC_COLORGAMUT* : CSC color gamut conversion standard format
- *ISP_CSC_MATRX_S* : CSC custom conversion matrix

38.3.1 ISP_CSC_ATTR_S

【Description】

CSC attribute parameter

【Syntax】

```
typedef struct _ISP_CSC_ATTR_S {
  CVI_BOOL Enable;
  ISP_CSC_COLORGAMUT enColorGamut;
  CVI_U8 UpdateInterval;
  CVI_U8 Hue;
  CVI_U8 Luma;
  CVI_U8 Contrast;
  CVI_U8 Saturation;
  ISP_CSC_MATRX_S stUserMatrx;
} ISP_CSC_ATTR_S;
```

【Member】

Member	Description
Enable	CSC module enable Value range: [0, 1] Data type: CVI_BOOL
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
enColorGamut	Color gamut conversion type selection 0: ISP_CSC_COLORGAMUT_BT601 1: ISP_CSC_COLORGAMUT_BT709 2: ISP_CSC_COLORGAMUT_BT2020 3: ISP_CSC_COLORGAMUT_USER
Hue	Adjust chroma Value range: [0x0, 0x64] Data type: CVI_U8
Luma	adjust brightness Value range: [0x0, 0x64] Data type: CVI_U8
Contrast	adjust contrast Value range: [0x0, 0x64] Data type: CVI_U8
Saturation	adjust saturation Value range: [0x0, 0x64] Data type: CVI_U8
stUserMatrx	Custom gamut conversion matrix

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCscAttr*
- *CVI_ISP_GetCscAttr*

38.3.2 ISP_CSC_COLORGAMUT

【Description】

CSC color gamut conversion standard format

【Syntax】

```
typedef enum _ISP_CSC_COLORGAMUT {
 ISP_CSC_COLORGAMUT_BT601,
 ISP_CSC_COLORGAMUT_BT709,
 ISP_CSC_COLORGAMUT_BT2020,
 ISP_CSC_COLORGAMUT_USER,
```

(continues on next page)

(continued from previous page)

```
  ISP_CSC_COLORGAMUT_NUM
} ISP_CSC_COLORGAMUT;
```

【Member】

Member	Description
ISP_CSC_COLORGAMUT_BT601	BT601 Color Gamut Conversion Standard
ISP_CSC_COLORGAMUT_BT709	BT709 Color Gamut Conversion Standard
ISP_CSC_COLORGAMUT_BT2020	BT2020 Color Gamut Conversion Standard
ISP_CSC_COLORGAMUT_USER	Custom color gamut conversion, used in conjunction with ISP_CSC_MATRX_S

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCscAttr*
- *CVI_ISP_GetCscAttr*

38.3.3 ISP_CSC_MATRX_S

【Description】

CSC custom conversion matrix

【Syntax】

```
typedef struct _ISP_CSC_MATRX_S {
  CVI_S16 userCscCoef[CSC_MATRIX_SIZE];
  CVI_S16 userCscOffset[CSC_OFFSET_SIZE];
} ISP_CSC_MATRX_S;
```

【Member】

Member	Description
userCscCoef[CSC_MATRIX_SIZE]	Coefficients of 3*3 color gamut conversion matrix Value range: [-0x2000, 0x1fff] Data type: CVI_S16
userCscOffset[CSC_OFFSET_SIZE]	Output offset Value range: [-0x100, 0xff] Data type: CVI_S16

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetCscAttr*

- *CVI_ISP_GetCscAttr*

39 VC

39.1 Function Overview

Adjust the Motion Map settings on the Video Codec side

39.2 API Reference

- *CVI_ISP_SetVCAttr* : Set VC attribute parameter
- *CVI_ISP_GetVCAttr* : Get VC attribute parameter

39.2.1 CVI_ISP_SetVCAttr

【Description】

Set VC attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_SetVCAttr(VI_PIPE ViPipe, const ISP_VC_ATTR_S *pstVCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstVCAttr	VC attribute parameter	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_GetVCAttr*

39.2.2 CVI_ISP_GetVCAttr

【Description】

Get VC attribute parameter

【Syntax】

```
CVI_S32 CVI_ISP_GetVCAttr(VI_PIPE ViPipe, ISP_VC_ATTR_S *pstVCAttr);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstVCAttr	VC attribute parameter	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetVCAttr*

39.3 Data Types

- *ISP_VC_ATTR_S* : VC attribute parameter

39.3.1 ISP_VC_ATTR_S

【Description】

VC attribute parameter

【Syntax】

```
typedef struct _ISP_VC_ATTR_S {
    CVI_U8 UpdateInterval;
    CVI_U8 MotionThreshold;
} ISP_VC_ATTR_S;
```

【Member】

Member	Description
UpdateInterval	Affects the parameter update interval, the larger the value, the slower the screen changes and the better the performance Value range: [0x0, 0xff] Data type: CVI_U8
MotionThreshold	Motion Map Threshold Value range: [0x0, 0xff] Data type: CVI_U8

【Note】

None.

【Related Data Type and Interface】

- *CVI_ISP_SetVCAttr*
- *CVI_ISP_GetVCAttr*

40 Statistical information

40.1 Overview

This section describes the 3A statistics provided by ISP and the configuration method.

40.2 API Reference

The interface of statistical information must call CVI_ISP_Init interface.

- *CVI_ISP_SetStatisticsConfig* : Set ISP statistics configuration.
- *CVI_ISP_GetStatisticsConfig* : Get ISP statistics configuration.
- *CVI_ISP_GetAEStatistics* : Get ISP AE statistics information.
- *CVI_ISP_GetWBStatistics* : Get ISP AWB statistics information.
- *CVI_ISP_GetFocusStatistics* : Get ISP AF statistics information.

40.2.1 CVI_ISP_SetStatisticsConfig

【Description】

Set ISP statistics configuration.

【Syntax】

```
CVI_S32 CVI_ISP_SetStatisticsConfig(VI_PIPE ViPipe, const ISP_STATISTICS_CFG_S
↪*pstStatCfg);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstStatCfg	ISP statistics configuration	Input

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

```
// Set AE / AWB / AF related statistic window setting
VI_PIPE ViPipe = 0;
ISP_STATISTICS_CFG_S stsCfg;

// Set AE0 windows ROI.
stsCfg.stAECfg.stCrop[0].bEnable = 1;
stsCfg.stAECfg.stCrop[0].u16X = stsCfg.stAECfg.stCrop[0].u16Y = 0;
stsCfg.stAECfg.stCrop[0].u16W = 1920;
stsCfg.stAECfg.stCrop[0].u16H = 1080;

// Set the window number of AWB statistic value in x & y direction.
stsCfg.stWBCfg.u16ZoneRow = AWB_ZONE_ORIG_ROW;
stsCfg.stWBCfg.u16ZoneCol = AWB_ZONE_ORIG_COLUMN;

// Set AWB window ROI.
stsCfg.stWBCfg.stCrop.u16X = stsCfg.stWBCfg.stCrop.u16Y = 0;
stsCfg.stWBCfg.stCrop.u16W = 1920;
stsCfg.stWBCfg.stCrop.u16H = 1080;

// Set AWB statistical brightness threshold.
stsCfg.stWBCfg.u16BlackLevel = 0;
stsCfg.stWBCfg.u16WhiteLevel = 4095;

// Set the AF statistics switch.
stsCfg.stFocusCfg.stConfig.bEnable = 1;

// Set the preprocessing switch of AF statistics.
stsCfg.stFocusCfg.stConfig.stRawCfg.PreGammaEn = 0;
stsCfg.stFocusCfg.stConfig.stPreFltCfg.PreFltEn = 1;

// Set the window number of AF statistic value in x & y direction.
stsCfg.stFocusCfg.stConfig.u16Hwnd = 17;
stsCfg.stFocusCfg.stConfig.u16Vwnd = 15;
```

(continues on next page)

(continued from previous page)

```
// Set AF window ROI.
stsCfg.stFocusCfg.stConfig.stCrop.bEnable = 1;
stsCfg.stFocusCfg.stConfig.stCrop.u16X = AF_XOFFSET_MIN;
stsCfg.stFocusCfg.stConfig.stCrop.u16Y = AF_YOFFSET_MIN;
stsCfg.stFocusCfg.stConfig.stCrop.u16W = 1920 - AF_XOFFSET_MIN * 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16H = 1080 - AF_YOFFSET_MIN * 2;

// Set the AF horizontal low pass filter coefficient.
stsCfg.stFocusCfg.stConfig.u8HFltShift = 1;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[0] = 1;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[1] = 2;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[2] = 3;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[3] = 5;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[4] = 10;

// Set the AF horizontal high pass filter coefficient.
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[1] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[2] = 13;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[3] = 24;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[4] = 0;

// Set the AF vertical high pass filter coefficient.
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[0] = 13;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[1] = 24;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[2] = 0;

// Set the value switch of each statistical value.
stsCfg.unKey.bit1FEAeGloStat = stsCfg.unKey.bit1FEAeLocStat =
stsCfg.unKey.bit1AwbStat1 = stsCfg.unKey.bit1AwbStat2 = stsCfg.unKey.
↪bit1FEAfStat = 1;
CVI_ISP_SetStatisticsConfig(ViPipe, &stsCfg);

// Set the AF low pass filter coefficient rule.
AF low pass filter has nine parameters {x0, x1, x2, x3, x4, x5, x6, x7, x8} .␣
↪The last four parameters are symmetrical with the first four parameters (x0 =␣
↪x8, x1 = x7, x2 = x6, x3 = x5) . Users only need to set the first five␣
↪parameters (x0 ~ x5).

// Set the AF level high pass filter coefficient rule.
AF high pass filter has nine parameters {x0, x1, x2, x3, x4, x5, x6, x7, x8}.␣
↪The last four parameters are the same as the first four parameters, with the␣
↪same number but opposite sign: (x0 = -x8, x1 = -x7, x2 = -x6, x3 = -x5).␣
↪Users only need to set the first five parameters (x0 ~ x5).
// Set the AF vertical high pass filter coefficient rule.
AF horizontal high pass filter has five parameters: {x0, x1, X2, X3, X4}. The␣
↪last two parameters are the same as the first two parameters, but the␣
↪positive and negative signs are opposite: (x0 = - x4, X1 = - x3). Users only␣
↪need to set the first three parameters (x0 ~ x2).
```

filter example of parameter setting

the band through which the filter can pass	Filter parameters x0	Filter parameters x1	Filter parameters x2
0.1 ~ 0.2	20	16	0
0.1 ~ 0.5	17	20	0
0.1 ~ 0.6	13	24	0
0.1 ~ 0.7	12	25	0
0.1 ~ 0.8	-10	-27	0
0.2 ~ 0.9	-10	27	0

【Related Topic】

- *CVI_ISP_GetStatisticsConfig*

40.2.2 CVI_ISP_GetStatisticsConfig

【Description】

Get ISP statistics configuration.

【Syntax】

```
CVI_S32 CVI_ISP_GetStatisticsConfig(VI_PIPE ViPipe, ISP_STATISTICS_CFG_S
↪*pstStatCfg);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstStatCfg	ISP statistics configuration	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

- *CVI_ISP_SetStatisticsConfig*

40.2.3 CVI_ISP_GetAEStatistics

【Description】

Get ISP AE statistics information.

【Syntax】

```
CVI_S32 CVI_ISP_GetAEStatistics(VI_PIPE ViPipe, ISP_AE_STATISTICS_S *pstAeStat);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstAeStat	AE statistics	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

```
//Configure AE window weight
#define AE_ZONE_ROW 15
#define AE_ZONE_COLUMN 17

ISP_AE_WIN_STATISTICS_CFG_S  aeWinCfg;
CVI_U8 i,j;
VI_PIPE ViPipe = 0;
CVI_U8 u8Weighttable[AE_ZONE_ROW][ AE_ZONE_COLUMN]={
{ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 },
{ 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1 },
{ 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1 },
```

(continues on next page)

(continued from previous page)

```
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 8, 8, 8, 8, 8, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 1 },
{ 1, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 1 },
{ 1, 2, 2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 1 },
{ 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1 },};

CVI_ISP_GetAEWinStatistics(ViPipe, &aeWinCfg);
for (i = 0; i < AE_ZONE_ROW; i++)
{
    for (j = 0; j < AE_ZONE_COLUMN; j++)
    {
        aeWinCfg.au8Weight[i][j] = u8Weighttable[i][j];
    }
}
CVI_ISP_SetAEWinStatistics(ViPipe, &aeWinCfg);
```

【Related Topic】

None.

40.2.4 CVI_ISP_GetWBStatistics

【Description】

Get ISP AWB statistics information.

【Syntax】

```
CVI_S32 CVI_ISP_GetWBStatistics(VI_PIPE ViPipe, ISP_WB_STATISTICS_S
↪*pstAwbStat);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstAwbStat	AWB statistics	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

None.

40.2.5 CVI_ISP_GetFocusStatistics

【Description】

Get ISP AF statistics information.

【Syntax】

```
CVI_S32 CVI_ISP_GetFocusStatistics(VI_PIPE ViPipe, ISP_AF_STATISTICS_S
*pstAfStat);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstAfStat	AF statistics	Output

【Return Value】

Return Value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

None.

40.3 Data Types

40.3.1 ISP_STATISTICS_CTRL_U

【Description】

Define ISP statistics information enablement.

【Syntax】

```
typedef union _ISP_STATISTICS_CTRL_U {
  CVI_U64 u64Key;
  struct {
    CVI_U64 bit1FEAeGloStat : 1; /* [0] */
    CVI_U64 bit1FEAeLocStat : 1; /* [1] */
    CVI_U64 bit1AwbStat1 : 1; /* [2] */
    CVI_U64 bit1AwbStat2 : 1; /* [3] */
    CVI_U64 bit1FEAfStat : 1; /* [4] */
    CVI_U64 bit14Rsv : 59; /* [5:63] */
} ISP_STATISTICS_CTRL_U;
```

【Member】

Member	Description
bit1FEAeGloStat	AE global statistics enable
bit1FEAeLocStat	AE block statistics enable
bit1AwbStat1	AWB global statistics enable
bit1AwbStat2	AWB block statistics enable
bit1FEAfStat	AF block statistics enable
bit14Rsv	Reserved bit

【Note】

None.

【Related Data Type and Interface】

None.

40.3.2 ISP_AE_STATISTICS_CFG_S

【Description】

Configure AE statistics.

【Syntax】

```
typedef struct _ISP_AE_STATISTICS_CFG_S {
    CVI_BOOL bHisStatisticsEnable; /*RW; Range:[0x0,0x1]*/
    ISP_AE_CROP_S stCrop[AE_MAX_NUM];
    ISP_AE_FACE_CROP_S stFaceCrop[FACE_WIN_NUM];
    CVI_BOOL fast2A_ena; /*RW; Range:[0x0,0x1]*/
    CVI_U8 fast2A_ae_low; /*RW; Range:[0x0,0xFF]*/
    CVI_U8 fast2A_ae_high; /*RW; Range:[0x0,0xFF]*/
    CVI_U16 fast2A_awb_top; /*RW; Range:[0x0,0xFFF]*/
    CVI_U16 fast2A_awb_bot; /*RW; Range:[0x0,0xFFF]*/
    CVI_U16 over_exp_thr; /*RW; Range:[0x0,0x3FF]*/
    CVI_U8 au8Weight[AE_WEIGHT_ZONE_ROW][AE_WEIGHT_ZONE_COLUMN]; /*RW;
↪Range:[0x0, 0xF]*/
} ISP_AE_STATISTICS_CFG_S;
```

【Member】

Member	Description
enAESwitch	HiSilicon compatible parameter, currently not used
stHistConfig	HiSilicon compatible parameter, currently not used
enFourPlaneMode	HiSilicon compatible parameter, currently not used
enHistMode	HiSilicon compatible parameter, currently not used
enAverMode	HiSilicon compatible parameter, currently not used
enMaxGainMode	HiSilicon compatible parameter, currently not used
stCrop[AE_MAX_NUM];	AE statistical value input image cropping setting
au8Weight[AE_ZONE_ROW][AE_ZONE_COLUMN]	AE window 17x15 weight setting
bHisStatisticsEnable	AE/Hist sts enable
u8StatisticsShiftBits	The AE statistics should be scaled down by a few bits
u16RGain	Whether the R channel passes through RGain
u16GGain	Whether the G channel passes through GGain
u16BGain	Whether the B channel passes through BGain

【Note】

None.

【Related Data Type and Interface】

- CVI_ISP_SetAEWinStatistics

40.3.3 ISP_AE_CROP_S

【Description】

Set AE statistics input image reduction.

【Syntax】

```
typedef struct _ ISP_AE_CROP_S {
  CVI_BOOL bEnable;
  CVI_U16 u16X;
  CVI_U16 u16Y;
  CVI_U16 u16W;
  CVI_U16 u16H;
} ISP_AE_CROP_S;
```

【Member】

Member	Description
bEnable	Enable Cro; 0 is not enabled, the whole picture will be set as the clipping setting; 1 is enabled
u16X	Start position of CropX, the value should be greater than 0
u16Y	Start position of CropY, the value should be greater than 0
u16W	Crop width
u16H	Crop height

【Note】

None.

【Related Data Type and Interface】

None.

40.3.4 ISP_AE_FACE_CROP_S

【Description】

Set face AE statistics input image reduction.

【Syntax】

```
typedef struct _ISP_AE_FACE_CROP_S {
  CVI_BOOL bEnable;
  CVI_U16 u16X;
  CVI_U16 u16Y;
  CVI_U16 u16W;
  CVI_U16 u16H;
} ISP_AE_FACE_CROP_S;
```

【Member】

Member	Description
bEnable	Enable Cro; 0 is not enabled, the whole picture will be set as the clipping setting; 1 is enabled
u16X	Start position of CropX, the value should be greater than 0
u16Y	Start position of CropY, the value should be greater than 0
u16W	Crop width
u16H	Crop height

【Note】

None.

【Related Data Type and Interface】

None.

40.3.5 ISP_WB_STATISTICS_CFG_S

【Description】

Configure WB statistics.

【Syntax】

```
typedef struct _ISP_WB_STATISTICS_CFG_S {
  ISP_AWB_SWITCH_E enAWBSwitch;
  CVI_U16 u16ZoneRow;
  CVI_U16 u16ZoneCol;
  CVI_U16 u16ZoneBin;
  CVI_U16 au16HistBinThresh[4];
  CVI_U16 u16WhiteLevel;
  CVI_U16 u16BlackLevel;
  CVI_U16 u16CbMax;
  CVI_U16 u16CbMin;
  CVI_U16 u16CrMax;
  CVI_U16 u16CrMin;
  ISP_AWB_CROP_S stCrop;
} ISP_WB_STATISTICS_CFG_S;
```

【Member】

Member	Description
enAWBSwitch	ISP_AWB_AFTER_DG = 0 ISP_AWB_AFTER_Expander, ISP_AWB_AFTER_DRC, ISP_AWB_SWITCH_BUTT Set AWB block location
u16ZoneRow	AWB statistics window rows
u16ZoneCol	AWB statistics window columns
u16ZoneBin	HiSilicon compatible parameter, currently not used
au16HistBinThresh	HiSilicon compatible parameter, currently not used
u16WhiteLevel	Threshold setting of AWB highlight exclusion
u16BlackLevel	Threshold setting of AWB dark spot exclusion
u16CbMax	The maximum value of R/G under different ISO
u16CbMin	The maximum value of R/G under different ISO
u16CrMax	The maximum value of R/G under different ISO
u16CrMin	The maximum value of R/G under different ISO

【Note】

None.

【Related Data Type and Interface】

None.

40.3.6 ISP_AWB_CROP_S

【Description】

Set AWB statistics input image reduction.

【Syntax】

```
typedef struct _ISP_ AWB_CROP_S {
  CVI_BOOL bEnable;
  CVI_U16 u16X;
  CVI_U16 u16Y;
  CVI_U16 u16W;
  CVI_U16 u16H;
} ISP_ AWB_CROP_S;
```

【Member】

Member	Description
bEnable	Enable Cro; 0 is not enabled, the whole picture will be set as the clipping setting; 1 is enabled
u16X	Start position of CropX, the value should be greater than 0
u16Y	Start position of CropY, the value should be greater than 0
u16W	Crop width Value range: [16 * u16ZoneCol, image width]
u16H	Crop height Value range: [0, image width - 16 * u16ZoneCol]

【Note】

None.

【Related Data Type and Interface】

None.

40.3.7 ISP_WB_STATISTICS_S

【Description】

Define AWB statistics.

【Syntax】

```
typedef struct _ISP_WB_STATISTICS_S {
  CVI_U16 u16GlobalR;
  CVI_U16 u16GlobalG;
  CVI_U16 u16GlobalB;
  CVI_U16 u16CountAll;
  CVI_U16 au16ZoneAvgR[AWB_ZONE_NUM];
  CVI_U16 au16ZoneAvgG[AWB_ZONE_NUM];
  CVI_U16 au16ZoneAvgB[AWB_ZONE_NUM];
  CVI_U16 au16ZoneCountAll[AWB_ZONE_NUM];
  ISP_AWB_GRID_INFO_S stGridInfo;
} ISP_WB_STATISTICS_S;

Macro Definition
#define AWB_ZONE_ORIG_ROW (32)
#define AWB_ZONE_ORIG_COLUMN (64)
#define AWB_ZONE_NUM (AWB_ZONE_ORIG_ROW * AWB_ZONE_ORIG_COLUMN)
```

【Member】

Member	Description
u16GlobalR	The average value of global R component in Bayer domain. Value range: [0, 0x3ff] Data type: CVI_BOOL
u16GlobalG	The average value of global G component in Bayer domain Value range: [0, 0x3ff] Data type: CVI_BOOL
u16GlobalB	The average value of global B component in Bayer domain Value range: [0, 0x3ff] Data type: CVI_BOOL
u16CountAll	The number of pixels in the global statistical region Value range: [0, 0x3ff] Data type: CVI_BOOL
au16ZoneAvgR	The average value of R component in Bayer domain Value range: [0, 0x3ff] Data type: CVI_BOOL
au16ZoneAvgG	The average value of G component in Bayer domain Value range: [0, 0x3ff] Data type: CVI_BOOL
au16ZoneAvgB	The average value of B component in Bayer domain Value range: [0, 0x3ff] Data type: CVI_BOOL
au16ZoneCountAll	The number of pixels between Bayer domains Value range: [0, 0xfff] Data type: CVI_BOOL
ISP_AWB_GRID_INFO_S	AWB statistics coordinate information

【Note】

None,

【Related Data Type and Interface】

None.

40.3.8 ISP_AWB_GRID_INFO_S

【Description】

AWB Statistics Coordinate Information

【Syntax】

```
typedef struct _ISP_AWB_GRID_INFO_S {
  CVI_U16 au16GridYPos[AWB_ZONE_ORIG_ROW + 1];
  CVI_U16 au16GridXPos[AWB_ZONE_ORIG_COLUMN + 1];
  CVI_U8 u8Status;
} ISP_AWB_GRID_INFO_S;

Macro Definition
```

(continues on next page)

(continued from previous page)

```
#define AWB_ZONE_ORIG_ROW (32)
#define AWB_ZONE_ORIG_COLUMN (32)
```

【Member】

Member	Description
au16GridYPos	Not used yet
au16GridXPos	Not used yet
u8Status	Not used yet

【Note】

None.

【Related Data Type and Interface】

None.

40.3.9 ISP_FOCUS_STATISTICS_CFG_S

【Description】

AF statistics configuration

【Syntax】

```
typedef struct _ISP_FOCUS_STATISTICS_CFG_S {
  ISP_AF_CFG_S stConfig;
  ISP_AF_H_PARAM_S stHParam_FIR0;
  ISP_AF_H_PARAM_S stHParam_FIR1;
  ISP_AF_V_PARAM_S stVParam_FIR;
} ISP_FOCUS_STATISTICS_CFG_S;
```

【Member】

Member	Description
stConfig	AF global configuration parameter
stHParam_FIR0	Horizontal filter first set of FIR parameter settings
stHParam_FIR1	Horizontal filter second set of FIR parameter settings
stVParam_FIR	FIR parameter setting of vertical filter

【Note】

None.

【Related Data Type and Interface】

None.

40.3.10 ISP_AF_CFG_S

【Description】

Define AF statistics parameter configuration

【Syntax】

```
typedef struct _ISP_AF_CFG_S {
    CVI_BOOL bEnable;
    CVI_U16 u16Hwnd; /*RW; Range:[0x2, 0x11]*/
    CVI_U16 u16Vwnd; /*RW; Range:[0x2, 0xF]*/
    CVI_U8 u8HFltShift; /*RW; Range:[0x0, 0xF]*/
    CVI_S8 s8HVFltLpCoeff[FIR_H_GAIN_NUM]; /*RW; Range:[0x0, 0x1F]*/
    ISP_AF_RAW_CFG_S stRawCfg;
    ISP_AF_PRE_FILTER_CFG_S stPreFltCfg;
    ISP_AF_CROP_S stCrop;
    CVI_U8 H0FltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 H1FltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 V0FltCoring; /*RW; Range:[0x0, 0xFF]*/
    CVI_U16 u16HighLumaTh; /*RW; Range:[0x0, 0xFF]*/
    CVI_U8 u8ThLow;
    CVI_U8 u8ThHigh;
    CVI_U8 u8GainLow; /*RW; Range:[0x0, 0xFE]*/
    CVI_U8 u8GainHigh; /*RW; Range:[0x0, 0xFE]*/
    CVI_U8 u8SlopLow; /*RW; Range:[0x0, 0xF]*/
    CVI_U8 u8SlopHigh; /*RW; Range:[0x0, 0xF]*/
} ISP_AF_CFG_S;
```

【Member】

Member	Description
bEnable	AF is enabled, 0 is not enabled, 1 is enabled
u16Hwnd	The maximum number of AF horizontal windows can be set to 17
u16Vwnd	The maximum number of AF vertical windows can be set to 15
u8HFltShift	AF low pass filter statistics shift register values Value range : [0, 15]
s8HVFltLpCoeff[FIR_H_GAIN_NUM]	AF low pass filter coefficient is used to control the frequency response of FIR filter Value range : [-32, 31]
stRawCfg	AF Bayer domain related configuration
stPreFltCfg	AF prefilter processing enable setting
stCrop	Clipping configuration of AF input image
u16HighLumaTh	AF highlight statistics threshold setting Value range : [0, 4095]

【Note】

None.

【Related Data Type and Interface】

None.

40.3.11 ISP_AF_RAW_CFG_S

【Description】

AF Bayer Domain Image Preprocessing Configuration

【Syntax】

```
typedef struct _ISP_AF_RAW_CFG_S {
    CVI_U8 PreGammaEn;
    CVI_U8 PreGammaTable[AF_GAMMA_NUM];
} ISP_AF_RAW_CFG_S;
```

【Member】

Member	Description
PreGammaEn	AF module Gamma enabled, 0 not enabled, 1 enabled
PreGammaTable[AF_GAMMA_NUM]	AF module gamma table settings

【Note】

None.

【Related Data Type and Interface】

None.

40.3.12 ISP_AF_PRE_FILTER_CFG_S

【Description】

AF Bayer Domain Image Filtering Preprocessing Configuration

【Syntax】

```
typedef struct _ISP_AF_PRE_FILTER_CFG_S {
    CVI_BOOL PreFltEn;
} ISP_AF_PRE_FILTER_CFG_S;
```

【Member】

Member	Description
PreFltEn	AF module filter preprocessing enabled, 0 not enabled, 1 enabled

【Note】

None.

【Related Data Type and Interface】

None.

40.3.13 ISP_AF_CROP_S

【Description】

AF Statistics Input Image Clipping Settings

【Syntax】

```
typedef struct _ISP_ AF _CROP_S {
  CVI_BOOL bEnable;
  CVI_U16 u16X;
  CVI_U16 u16Y;
  CVI_U16 u16W;
  CVI_U16 u16H;
} ISP_AF_CROP_S;
```

【Member】

Member	Description
bEnable	Crop enabled, 0 enabled, whole picture set as crop setting, 1 enabled
u16X	CropX start position, Value range:[]
u16Y	CropY start position, Value range:[]
u16W	Crop width Value range: [16 * u16ZoneCol, image width - 16]
u16H	Crop height Value range: [16 * u16ZoneRow, image height - 4]

【Note】

None.

【Related Data Type and Interface】

None.

40.3.14 ISP_AF_H_PARAM_S

【Description】

Define AF horizontal filter parameter settings

【Syntax】

```
typedef struct _ISP_AF_H_PARAM_S {
  CVI_S8 s8HFltHpCoeff[FIR_H_GAIN_NUM];
} ISP_AF_H_PARAM_S;
```

【Member】

Member	Description
s8HFltHpCoeff[FIR_H_GAIN_NUM]	AF horizontal filter factor, used to control the frequency response of FIR filters value range : [-32, 31]

【Note】

None.

【Related Data Type and Interface】

None.

40.3.15 ISP_AF_V_PARAM_S

【Description】

Define AF horizontal filter parameter settings

【Syntax】

```
typedef struct _ISP_AF_V_PARAM_S {
  CVI_S8 s8VFltHpCoeff[FIR_V_GAIN_NUM];
} ISP_AF_V_PARAM_S;
```

【Member】

Member	Description
s8VFltHpCoeff[FIR_V_GAIN_NUM]	AF vertical filter factor, used to control the frequency response of FIR filters value range : [-32, 31]

【Note】

None.

【Related Data Type and Interface】

None.

40.3.16 ISP_STATISTICS_CFG_S

【Description】

ISP Statistics Message Configuration

【Syntax】

```
typedef struct _ISP_STATISTICS_CFG_S {
  ISP_STATISTICS_CTRL_U unKey;
  ISP_AE_STATISTICS_CFG_S stAECfg;
  ISP_WB_STATISTICS_CFG_S stWBCfg;
  ISP_FOCUS_STATISTICS_CFG_S stFocusCfg;
} ISP_STATISTICS_CFG_S;
```

【Member】

Member	Description
unKey	Statistical Enabling
stAECfg	AE Statistics Configuration
stWBCfg	WB Statistics Configuration
stFocusCfg	AF Statistics Configuration

【Note】

None.

【Related Data Type and Interface】

None.

40.3.17 ISP_FOCUS_ZONE_S

【Description】

Define the statistical results calculated by AF

【Syntax】

```
typedef struct _ISP_FOCUS_ZONE_S {
  CVI_U16 u16HlCnt;
  CVI_U64 u64h0;
  CVI_U64 u64h1;
  CVI_U32 u32v0;
} ISP_FOCUS_ZONE_S;
```

【Member】

Member	Description
u16HlCnt	The number of points exceeding the highlight threshold is counted within the AF zones, and the entire picture will be set as the crop setting, 1 being enabled
u64h0	Statistical results of the first group of FIR filters in the horizontal direction within the AF partition interval
u64h1	Statistical results of the second group of FIR filters in the horizontal direction within the AF partition interval
u32v0	Results of Statistical Vertical FIR Filters Between AF Partitions

【Note】

None.

【Related Data Type and Interface】

None.

40.3.18 ISP_FE_FOCUS_STATISTICS_S

【Description】

Define statistical information provided by AF FE

【Syntax】

```
typedef struct _ISP_FE_FOCUS_STATISTICS_S {
  ISP_FOCUS_ZONE_S stZoneMetrics[AF_ZONE_ROW][AF_ZONE_COLUMN];
} ISP_FE_FOCUS_STATISTICS_S;
```

【Member】

Member	Description
stZoneMetrics	Blocking Statistics for ISP AF

【Note】

None.

【Related Data Type and Interface】

None.

40.3.19 ISP_AF_STATISTICS_S

【Description】

Define all statistics provided by AF

【Syntax】

```
typedef struct _ISP_AF_STATISTICS_S {
  ISP_FE_FOCUS_STATISTICS_S stFEAFStat;
} ISP_AF_STATISTICS_S;
```

【Member】

Member	Description
stFEAFStat	Statistics of AF in Bayer Domain

【Note】

None.

【Related Data Type and Interface】

None.

41 Query internal status messages

41.1 Funtion Overview

This section describes the interfaces associated with Inner State Information.The purpose of this interface is to provide users with the true values currently set for querying the internal state of the system and for several ISO-related parameter.

Users can use this interface during debugging to obtain the actual values currently filled in by ISO-related parameter to verify that they are configured correctly.Only numeric values can be obtained through this interface, and related parameter cannot be changed.

41.2 API Reference

- *CVI_ISP_QueryInnerStateInfo* : Get the internal information of the system and the real values currently set for ISO-related parameter.

41.2.1 CVI_ISP_QueryInnerStateInfo

【Description】

Get internal information about the system and the real values currently set for ISO-related parameter

【Syntax】

```
CVI_S32 CVI_ISP_QueryInnerStateInfo(VI_PIPE ViPipe, ISP_INNER_STATE_INFO_S
↪*pstInnerStateInfo);
```

【Parameter】

Parameter	Description	Input/Output
ViPipe	VI_PIPE number	Input
pstInnerStateInfo	Internal Information and Current Actual Parameter Settings	Output

【Return Value】

return value	Description
0	Success
Non 0	Failure. An error code is returned. For details, see chapter *Error Codes*.

【Requirement】

- Header files: cvi_isp.h, cvi_comm_isp.h
- Library files: libisp.so

【Note】

None.

【Example】

None.

【Related Topic】

None.

41.3 Data Types

- *ISP_INNER_STATE_INFO_S* : Define internal information and current actual parameter settings

41.3.1 ISP_INNER_STATE_INFO_S

【Description】

Define internal information and current actual parameter settings

【Syntax】

```
#define MAX_HIST_BINS 256
#define MAX_EXPOSURE_RATIO 256
#define LTM_DARK_CURVE_NODE_NUM 257
#define LTM_BRIGHT_CURVE_NODE_NUM 513

typedef struct _ISP_INNER_STATE_INFO_S {
  CVI_U32 blcOffsetR;
  CVI_U32 blcOffsetGr;
  CVI_U32 blcOffsetGb;
  CVI_U32 blcOffsetB;
  CVI_U32 blcGainR;
  CVI_U32 blcGainGr;
  CVI_U32 blcGainGb;
  CVI_U32 blcGainB;
  CVI_S32 ccm[9];
```

(continues on next page)

(continued from previous page)

```
    CVI_U16 drcGlobalToneBinNum;
    CVI_U16 drcGlobalToneBinSEStep;
    CVI_U32 drcGlobalTone[LTM_GLOBAL_CURVE_NODE_NUM];
    CVI_U32 drcDarkTone[LTM_DARK_CURVE_NODE_NUM];
    CVI_U32 drcBrightTone[LTM_BRIGHT_CURVE_NODE_NUM];
    CVI_BOOL bWDRSwitchFinish;
    // For 2TO1/3TO1/4TO1 use.
    CVI_U32 u32WDRExpRatioActual[ISP_WDR_FRAME_IDX_SIZE];
    ISP_MESH_SHADING_GAIN_LUT_S mlscGainTable;
} ISP_INNER_STATE_INFO_S;
```

【Member】

Member	Description
wdrHistBinNum	Number of array in the current wdr distribution histogram of the picture
blcOffsetR	Current value of black level subtracted from component R
blcOffsetGr	*Current value of black level subtracted from component Gr*
blcOffsetGb	*Current value of black level subtracted from component Gb*
blcOffsetB	Current value of black level subtracted from component B
blcGainR	Digital gain of current ISP in R component
blcGainGr	Digital gain of current ISP in Gr component
blcGainGb	Digital gain of current ISP in Gb component
blcGainB	Digital gain of current ISP in B component
Ccm[9]	True values of the color restore matrix currently used by ISP
wdrHistogramBefore[MAX_HIST_BINS * MAX_EXPOSURE_RATIO]	Distribution histogram of current wdr
wdrHistogramAfter[MAX_HIST_BINS * MAX_EXPOSURE_RATIO]	Hue mapping curves from the current wdr distribution histogram
drcDarkTone[LTM_DARK_CURVE_NODE_NUM]	Dark area hue mapping curve currently used by ISP
drcBrightTone[LTM_BRIGHT_CURVE_NODE_NUM]	Bright area hue mapping curve currently used by ISP
bWDRSwitchFinish	Indicate if sensor mode switch is complete

【Note】

None.

【Related Data Type and Interface】

None.

42 Debug

43 Error Codes

44 Proc Debugging Information Description

44.1 Function Overview

The debugging information uses the proc file system, which can reflect the set parameters, 3A statistics, 3A operation results and other information in the ISP module in real time according to the proc param and proc level set by the upper layer for developers to locate and analyze problems.

44.2 Usage Method

[File directory]

/proc/cvitek/isp

[Opening Method]

After calling CVI_ISP_After MemInit, call CVI_ISP_SetCtrlParam to set the ISP control parameter u32ProcParam=n, where n cannot be zero.

n is the frequency at which ISP information is collected. ISP information is collected every n frames. The default value is 30

Call CVI_ISP_SetCtrlParam to set the ISP control parameter u32ProcLevel=m, where m ranges from [0, 3]

- m=0 means proc function is off,
- m=1 means level1 level information is printed,
- m=2 means level2 level information is printed,
- m=3 means level3 level information is printed, the higher level3 level information is printed, and the higher level print content must contain lower level print content, default value is 0
- When using ISP_TOOL_DAEMON to turn on proc , you can Export PROC_LEVEL=n (n 1-3, n=0 does not open) before running ISP_TOOL_DAEMON

[View Information Methods]

After ISP module is running properly

By calling cat/proc/cvitek/isp in the console, the associated logs are then printed to the console cp/proc/cvitek/isp view to pc

44.3 ISP

44.3.1 LEVEL1 Debugging Information Analysis

Printing of ISP Module parameters

[parameters Description]

Parameter		Description
MODULE/CTRLPARAM	ProcParam	Indicates the collection frequency of collecting ISP information
	ProcLevel	Indicates the level level of ISP proc
	AEStatIntvl	Indicates the update frequency of ISP AE statistics
	AWBStatIntvl	Indicates the update frequency of ISP AWB statistics
	AFStatIntvl	Indicates the update frequency of ISP AF statistics
	UpdatePos	Indicates whether the interrupt update position is interrupted at the start of the frame or at the end of the frame, 0 is the start of the frame
	IntTimeOut	Indicates the maximum time to get the ISP interrupt timeout
	PwmNumber	Indicates pwm usage
	PortIntDelay	Indicates the Port interrupt delay time
FSWDR	Enable	WDR module enable
	MotionCompEnable	WDR motion detect enable switch
	CombineSNRAwareEn	The wide dynamic fusion mode with SNR perception is enabled, which can adaptively determine the fusion ratio of long and short frames according to the noise of short frames
	CombineSNRAwareLowThr	Short frame noise SNR adaptive low threshold. When the estimated noise of the short frame is lower than the low threshold value, general long and short frame fusion is performed, and the noise is between the low threshold value and the high threshold value, and the SNR adaptive fusion is performed in proportion.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	CombineSNRAware-HighThr	When the short frame estimation noise is higher than the high threshold value according to The SN RAwareToleranceLevel intensity performs SNR adaptive fusion, and the noise is between the low critical value and the high critical value, and the SNR adaptive fusion is performed in proportion.
	CombineS-NRAwareSmLevel	The smoothness of time domain changes in short frame noise SNR adaptation
	ExposureRatio	Exposure ratio of WDR long and short exposure
	ShortMaxVal	The maximum value of WDR wide dynamic range
	isManualMode	manual mode or automatic mode
	CombineLongThr	Long exposure threshold, image data below this threshold will only select long exposure data to synthesize WDR images
	CombineShortThr	Short exposure critical value, image data exceeding this critical value will only select short exposure data to synthesize WDR image
	CombineMinWeight	The minimum weight value for long and short exposure image data fusion. The larger the weight value, the more the proportion of long exposure in the fusion time, and vice versa, the more the proportion of short exposure
	CombineMaxWeight	Long and short exposure image data are fused with the highest weight value. The larger the weight value, the more the proportion of long exposure in the fusion time, and vice versa, the more the proportion of short exposure
	MergeMode	WDR motion detect mode 0: Motion detect information takes the maximum value of long and short frame motion information 1: Motion detect information is an equal proportion fusion of long and short frame motion information, and the ratio is defined by MergeModeAlpha
	MergeModeAlpha	MergeMode mode is 1 mobile information fusion ratio
	C ombineSNRAware-TolLevel	Noise Tolerance Strength of Short Frame Noise SNR Adaptation
Shading	MeshEnable	MeshLSC function enable
	MeshisManualMode	manual mode or automatic mode
	MeshStr	LSC compensation strength
	MeshLscGainLut Size	Color temperature adaptive LSC compensation gain table quantity

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	MeshLscGainLut-ColorTemp	The color temperature corresponding to the color temperature adaptive LSC compensation gain table
	RadialEnable	RadialLSC function enable
	RadialCenterX	X-direction coordinates of image sensor mirror center
	RadialCenterY	Y-direction coordinates of image sensor mirror center
	RadialStr	LSC compensation strength
	Low-RadialGainLut	LSC Radius Form Compensation Gain Table
	High-RadialGainLut	LSC Radius Form Compensation Gain Table
DCI	Enable	DCI module enable
	Speed	Smooth strength, the higher the value, the slower the change
	DciStrength	Used to control the strength of DCI, the larger the value, the greater the contrast.
	isManualMode	manual mode or automatic mode
	ContrastGain	Keep the lowest slope of each BIN, the larger the value, the smaller the slope
	BlcThr	The threshold value used to determine the range of the dark zone. The larger the value, the larger the range of dark areas included.
	WhtThr	The threshold used to determine the range of bright areas. The smaller the value, the larger the range of bright areas included.
	BlcCtrl	Used to determine the contrast of dark areas. When the value is 256, the contrast of dark areas remains unchanged. When it is larger than 256, the larger the value, the greater the contrast of the dark area; otherwise, the smaller the value, the smaller the contrast of the dark area.
	WhtCtrl	Used to determine the contrast of bright areas. When the value is 256, the contrast of the bright area remains unchanged. When the value is greater than 256, the larger the value, the greater the contrast of the bright area; on the contrary, the smaller the value, the smaller the contrast of the bright area.
Dehaze	Enable	Enable the Dehaze function
	isManualMode	manual mode or automatic mode
	Strength	Used to control the intensity of Dehaze. The larger the value, the stronger the defogging strength
BlackLevel	Enable	The BLC module enable
	isManualMode	manual mode or automatic mode
	PreOffsetR	BLC R pixel dark current value

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	PreOffsetGr	BLC GR pixel dark current value
	PreOffsetGb	BLC Gb pixel dark current value
	PreOffsetB	BLC B pixel dark current value
	PreGainR	BLC R pixel dark current compensation gain
	PreGainGr	BLC Gr pixel dark current compensation gain
	PreGainGb	BLC Gb pixel dark current compensation gain
	PreGainB	BLC B pixel dark current compensation gain
	PostOffsetR	BLC R pixel dark current value
	PostOffsetGr	BLC Gr pixel dark current value
	PostOffsetGb	BLC Gb pixel dark current value
	PostOffsetB	BLC B pixel dark current value
	PostGainR	BLC R pixel dark current compensation gain
	PostGainGr	BLC Gr pixel dark current compensation gain
	PostGainGb	BLC Gb pixel dark current compensation gain
	PostGainB	BLC B pixel dark current compensation gain
DPC	Enable	The DPC module enable
	isManualMode	manual mode or automatic mode
	ClusterSize	The upper limit of the clustered dead pixels area, the higher the value, the better the clustered dead pixels can be corrected, but it may cause the attenuation of the high-frequency area resolution
	BrightDefToNorPixRatio	The magnification of visible bright and bad pixel value and surrounding pixels
	DarkDefToNorPixRatio	The magnification of visible dark pixel value and surrounding pixels
	FlatThreR	The critical value of the R channel to determine the flat area, the smaller the value, the better the edge information can be preserved
	FlatThreG	The critical value of the B channel to determine the flat area, the smaller the value, the better the edge information can be preserved
	FlatThreB	The critical value of the B channel to determine the flat area, the smaller the value, the better the edge information can be preserved
	FlatThreMinG	The minimum critical value of the G channel to distinguish the flat area
	FlatThreMinRB	The minimum critical value of the RB channel to determine the flat area
TNR	Enable	TNR module enable
	DeflickerMode	Anti-flicker mode
	D eflickerToleranceLevel	Anti-flicker mode mode4 anti-flicker tolerance value, the larger the value, the better the anti-flicker, but the moving afterimage in the bright area will also increase.
	LowMtPrtEn	Airspace noise reduction micro motion protection enabled

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	isManualMode	manual mode or automatic mode
	TnrStrength0	Long Exposure TNR Intensity Gain
	MapThdLow0	Long Exposure TNR Intensity Upper Limit
	MapThdHigh0	Long exposure TNR intensity lower limit
	BrightnessNoiseLevelLE	Long Exposure Luminance Noise Tolerance Value
	BrightnessNoiseLevelSE	Short exposure luminance noise tolerance value
	RNoiseLevel0	Long exposure red channel noise tolerance value
	RNoiseHiLevel0	Long exposure red channel highlight noise tolerance value
	GNoiseLevel0	Long exposure green channel noise tolerance value
	GNoiseHiLevel0	Long exposure green channel highlight noise tolerance value
	BNoiseLevel0	Long exposure blue channel noise tolerance value
	BNoiseHiLevel0	Long exposure blue channel highlight noise tolerance value
	RNoiseLevel1	Short exposure red channel noise tolerance value
	RNoiseHiLevel1	Short exposure red channel highlight noise tolerance value
	GNoiseLevel1	Short exposure green channel noise tolerance value
	GNoiseHiLevel1	Short exposure green channel highlight noise tolerance value
	BNoiseLevel1	Short exposure blue channel noise tolerance value
	BNoiseHiLevel1	Short exposure blue channel highlight noise tolerance value
	LowMtPrtLevel	protection limit
	L2mIn0	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level.
	L2mOut0	Long-exposure TNR brightness versus intensity gain characteristics table. An array of four sets of values. Defines the intensity gain, the larger the value, the stronger the intensity.
	L2mIn1	Short exposure TNR luminance vs. intensity gain characteristics table. An array of four sets of values. Define the gray level, the larger the value, the higher the gray level.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	L2mOut1	Short exposure TNR luminance vs. intensity gain characteristics table. An array of four sets of values. Defines the intensity gain, the larger the value, the stronger the intensity.
	PrtctIn0	Long-exposure TNR momentum vs. smear removal degree characteristic table, an array consisting of four sets of values. Define the motion level to eliminate the tailing of moving objects, the larger the value, the stronger the motion range.
	PrtctOut0	Long-exposure TNR momentum vs. smear removal degree characteristic table, an array consisting of four sets of values. Define the motion level to remove the tailing of moving objects, the smaller the value, the stronger the tailing removal.
	PrtctIn1	Short-exposure TNR momentum vs. smear removal characteristic table, an array consisting of four groups of values. Define the motion level to eliminate the tailing of moving objects, the larger the value, the stronger the motion range.
	PrtctOut1	Short-exposure TNR momentum vs. smear removal characteristic table, an array consisting of four groups of values. Define the motion level to remove the tailing of moving objects, the smaller the value, the stronger the tailing removal.
	LowMtPrtIn	Define exercise class
	LowMtPrtOut	The larger the value, the less obvious the smear and the more obvious the noise
CAC	Enable	CAC module enable
	VarThr	Threshold for edge detect. The smaller the value, the more regions are judged as edges.
	PurpleDetRange	Threshold for purple fringing detect. The larger the value, the more areas are judged as purple fringing.
	PurpleCb	The coordinates of purple in the Cb domain.
	PurpleCr	The coordinates of purple in the Cr domain.
	GreenCb	The coordinates of green in Cb domain.
	GreenCr	The coordinates of green in Cr domain.
	isManualMode	manual mode or automatic mode
	DePurpleStr	Threshold for purple fringing detect. The larger the value, the more areas are judged as purple fringing.
CNR	Enable	The CNR module is enabled.
	isManualMode	manual mode or automatic mode

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	CnrStr	Color noise denoising strength. The larger the value, the stronger the color noise denoising.
	NoiseSuppressStr	Color noise suppression strength. The larger the value, the stronger the color noise removal.
	NoiseSuppressGain	Color noise suppression strength gain. The smaller the value, the stronger the color noise denoising.
	FilterType	Color noise denoising filter strength. The larger the value, the stronger the color noise removal.
	MotionNrStr	Adjust the color noise denoising strength of the moving object area. Larger values result in less color noise in motion areas.
	DetailSmoothMode	Denoise detail smoothing function enabled
Sharpen	Enable	Y Sharpen module enable
	EdgeGain	Edge Enhancement parameters The larger the value, the stronger the edge sharpening
	TextureGain	Enhancement parameters for detail | textures The larger the value, the greater the detail texture sharpening intensity
	EdgeThr	The boundary threshold between edge and noise, greater than this value is considered as edge, and less than this value is considered as noise The larger the value, the fewer edges are enhanced; the smaller the value, the more edges are enhanced. When the exposure gain is larger, the value is recommended to be set larger
	TextureThr	The demarcation threshold between detail texture and noise, greater than this value is regarded as detail texture, and less than this value is regarded as noise
	LumaAdpCoringEn	Automatic Luminance Noise Rejection Threshold Switch
	LumaAdpGainEn	Brightness sharpening weights enabled
	DeltaAdpGainEn	sharpen weights enabled
	W drCoringCompensationEn	Luminance sharpening noise value compensation is enabled in wdr mode.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	Wdr CoringCompensationMode	Compensation mode for luminance sharpening noise value in wdr mode. 0: Compensate luminance sharpening noise according to WdrCoringHighThrd and WdrCoringLowthd. 1: Automatically compensate luminance sharpening noise according to DRC tone mapping curve.
	W drCoringToleranceLevel	Tolerance value for luma sharpening noise compensation in wdr mode. The smaller the value, the more obvious the effect of sharpening, but the easier it is to sharpen the noise. The larger the value is, the less likely it is to sharpen the noise, but the sharpening effect is less obvious.
	WdrCoringHighThr	High threshold for luma sharpening noise wdr mode compensation. Noise compensation is not performed for brightness below this threshold. If the brightness is higher than this critical value, noise compensation will be performed according to the value of WdrC oringToleranceLevel.
	WdrCoringLowThr	Brightness between WdrCoringLowThrd and WdrCoringHighThrd is compensated proportionally. Low threshold for luma sharpening noise wdr mode compensation. Noise compensation is not performed for brightness below this threshold. If the brightness is higher than this critical value, noise compensation will be performed according to the value of WdrC oringToleranceLevel. Brightness between WdrCoringLowThrd and WdrCoringHighThrd is compensated proportionally.
	isManualMode	manual mode or automatic mode
	EdgeFreq	Image Directional Edge Band Control
	TextureFreq	Image non-directional detail texture band control
	GlobalGain	Global sharpening weight
	OverShootThr	White edge sharpening upper limit
	UnderShootThr	Lower limit of black edge sharpening
	YNoiseLevel	Luminance sharpening noise value magnification, one time is 64. The larger the value is, the more the brightness sharpening noise is amplified, and vice versa.
	DeltaAdpGain	sharpening weight

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	LumaAdpCoring	Luminance sharpening noise values, detail texture or edge enhancement will exclude the enhancement contributed by this tolerance value. The smaller the value, the more obvious the effect of sharpening, but the easier it is to sharpen the noise. The larger the value is, the less likely it is to sharpen the noise, but the sharpening effect is less obvious.
	LumaAdpGain	Brightness Sharpening Weight
Saturation	isManualMode	manual mode or automatic mode
	Saturation	saturation
Gamma	Enable	Gamma function enable
	enCurveType	Gamma Curve Type
	isManualMode	manual mode or automatic mode
HSV	Enable	Saturation Tuning enabled
	isManualMode	manual mode or automatic mode
	SatCoringLinearTh	coring value for saturation
	SatCoringLinearLmt	Maximum output saturation
CCM	Enable	CCM module enable
	isManualMode	manual mode or automatic mode
	ISOActEnable	Enable CCM Bypass function under low illumination
	TempActEnable	The CCM Bypass function is enabled under high and low color temperatures.
	CCMTabNum	The number of currently configured CCM matrices
	CCM	Color Correction Matrix
	CCMTab	CCM matrix coefficients at different color temperatures
YNR	Enable	YNR module enable
	CoringParamEnable	Control whether to use manual coring
	isManualMode	manual mode or automatic mode
	WindowType	Denoising filtering locality degree. The smaller the value, the more localized the effect
	DetailSmoothMode	Denoise detail smoothing function enabled
	NoiseSuppressStr	Noise suppression strength. The larger the value, the greater the intensity of bright noise removal
	FilterType	Denoising filter strength. The larger the value, the greater the intensity of bright noise removal
	MotionThr	Object movement threshold. The smaller the value, the larger the range of the frame detected as motion.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	MotionNrPosGain	Adjusts the strength of denoising in areas greater than the motion threshold. The larger the value, the less noise is preserved.
	MotionNrNegGain	Adjusts the strength of denoising in areas less than the motion threshold. Smaller values preserve less noise.
	VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged.
	CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region.
	CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region.
	NonDirFiltStr	Adjusts the strength of denoising in areas less than the motion threshold. Smaller values preserve less noise.
	VhDirFiltStr	Threshold for detect edges. The larger the value, the fewer the number of edges judged.
	AaDirFiltStr	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region.
	NoiseCoring-BaseLuma	The brightness value of the motion zone.
	NoiseCoringBaseOff-set	The brightness noise tolerance value of the motion area, the judgment of the motion area is linked with the detecting the TNR motion area. The larger the value, the greater the denoising strength of the motion area.
	NoiseCoringAd-vLuma	Quiet area brightness value
	NoiseCoringAdvOff-set	Luminance noise tolerance value in the static area, the judgment of the static area is linked with the detecting the TNR motion area. The larger the value, the stronger the noise removal in the static area.
WDRExposureAttr	ExpRatioType	Quiet area brightness value Luminance noise tolerance value in the static area, the judgment of the static area is linked with the detect of the TNR motion area. The larger the value, the stronger the noise removal in the static area.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	ExpRatio	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, au32ExpRatio is invalid. When enExpRatioType is OP_TYPE_MANUAL, au32ExpRatio is Rewritable, indicating the expected value of the exposure ratio of two adjacent frames of multi-frame synthesis WDR. Value range: [0x40, 0xFFF]
	ExpRatioMax	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMax indicates the maximum value of the ratio of the exposure time of the longest frame to the shortest frame. u32ExpRatioMax is invalid when enExpRatioType is OP_TYPE_MANUAL. 6bit decimal precision, 0x40 means the exposure ratio is 1 times.
	ExpRatioMin	Only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, u32ExpRatioMin indicates the minimum value of the ratio of long frame exposure time to short frame exposure time. When enExpRatioType is OP_TYPE_MANUAL, u32ExpRatioMin is invalid. The format is unsigned 6.6bit fixed point, 0x40 means that the ratio of long frame exposure time to short frame exposure time is 1 times. The default value is 0x40. Value range: [0x40, u32ExpRatioMax]
	Tolerance	Exposure ratio tolerance value, only valid in two-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the larger the value, it means that the exposure ratio remains unchanged when the dynamic range of the scene changes within a certain range. The default value is 0xC. Value range: [0x0, 0xFF]

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	Speed	The automatic exposure ratio adjustment speed is only valid in the two-frame composite WDR mode. When enExpRatioType is OP_TYPE_AUTO, the larger the value, the faster the automatic exposure ratio adjustment. The default value is 0x20. Value range: [0x0, 0xFF]
	RatioBias	The exposure ratio deviation value is only valid in multi-frame synthesis WDR mode. When enExpRatioType is OP_TYPE_AUTO, the larger the value, the larger the automatic exposure ratio. The default value is 0x400, which means that the calculation result of the automatic exposure ratio algorithm will not be adjusted. The exposure ratio adjusted by this value will be limited by the maximum/minimum value of the exposure ratio. Value range: [0x0, 0xFFFF]
	SECompensation	Adjust the target brightness value of the short frame picture Value range: [0x0, 0xFF]
	SEHisThr	Calculate the exposure ratio threshold when the short frame exceeds the long frame (1x = 64)
	SE His255CntTargetDownThr	If the number of short frame histogram bin 255 is greater than this threshold, the target brightness of the short frame will be lowered
	SE-His255CntTargetUpThr	If the number of short frame histogram bin 255 is less than this threshold, the target brightness of the downgraded short frame will be restored to the original target brightness
	L EHisCntTarget-DownRatio	If the number of long frame histogram bin 255 is greater than this threshold, the target brightness of the long frame will be lowered
	LEHisCntTarge-tUpRatio	If the number of long frame histogram bin 255 is less than this threshold, the target brightness of the downgraded long frame will be restored to the original target brightness
	LEAdjustTargetMin	Long frame Target min of each LV 40,40, 40, 40, 40, 40, 45, 50, 60, 60, 60, 60, 60, 60, 60, 60
	LEAdjustTargetMax	Long frame Target max of each LV 50,50, 50, 50, 50, 50, 60, 70, 85, 100, 110, 110, 120, 120, 120, 120

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	SEAdjustTargetMin	Short frame target min for each LV 20,20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20
	SEAdjustTargetMax	Short frame target max for each LV 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60, 60
	DiffPixelNum	The pixel number threshold of long frame histogram bin 255, if this value is exceeded, the target brightness of long exposure will be reduced
	LELowBinThr	Only windows whose long-frame brightness is greater than this threshold can be added to AE long-frame photometry
	LEHighBinThr	Only windows whose long-frame brightness is less than this threshold can be added to AE long-frame metering
	SELowBinThr	Only windows whose short frame brightness is greater than this threshold can be added to AE short frame metering
	SEHighBinTh	Only windows whose short-frame brightness is less than this threshold can be added to AE short-frame metering
ExposureAttr	ByPass	AE module bypass function enable, the default is CVI_FALSE In FSWDR mode, it indicates the exposure time of the current shortest frame (VS).
	OpType	Automatic exposure or manual exposure switch, the default is OP_TYPE_AUTO.
	AERunInterval	The interval at which the AE algorithm runs, and the value range is [1, 255]. When the value is 1, it means that the AE algorithm is run every frame; When the value is 2, it means to run the AE algorithm every 2 frames, and so on. It is recommended not to set this value greater than 2, otherwise the AE adjustment speed will be affected. In WDR mode, it is recommended to set this value to 1, so that AE convergence will be smoother. This value defaults to 1. Linear mode does not care about this value.
	ExpTimeOpType	Manual exposure time enable, the default value is OP_TYPE_AUTO
	AGainOpType	Manual sensor analog gain enable, the default value is OP_TYPE_AUTO
	DGainOpType	Manual sensor digital gain enable, the default value is OP_TYPE_AUTO

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	ISPDGainOpType	Manual ISP digital gain, 10bit decimal precision, the default value is 0x400.
	ExpTime	Manual exposure time, in microseconds (us), the default value is 0x4000
	AGain	Manual sensor analog gain, 10bit decimal precision, the default value is 0x400.
	DGain	Manual sensor digital gain, 10bit decimal precision, the default value is 0x400.
	ISPDGain	Manual ISP digital gain, 10bit decimal precision, the default value is 0x400.
	ISONumOpType	ISO num manual/automatic mode switching
	UseISONum	In manual mode, use the ISO num type to control the gain
	ISONum	In manual mode, the set ISO num
	ExpTimeRangeMax	Exposure time range, set the maximum value in microseconds (us).
	ExpTimeRangeMin	Exposure time range, set the minimum value in microseconds (us).
	AGainRangeMax	Sensor analog gain range, set the maximum value, 10bit decimal precision.
	AGainRangeMin	Sensor analog gain range, set the minimum value, 10bit decimal precision.
	DGainRangeMax	Sensor digital gain range, set the maximum value, 10bit decimal precision.
	DGainRangeMin	Sensor digital gain range, set the minimum value, 10bit decimal precision.
	ISPDGainRangeMax	ISP digital gain range, set the maximum value, 10bit decimal precision.
	ISPDGainRangeMin	ISP digital gain range, set minimum value, 10bit decimal precision.
	SysGainRangeMin	System gain range, setting minimum value, 10bit decimal precision
	SysGainRangeMax	System gain range, maximum setting, 10bit decimal precision
	Speed	The speed at which auto exposure is adjusted.
	BlackSpeedBias	Screen from dark to bright AE adjusts the deviation of the speed, the larger the value, the faster the screen from dark to bright.
	Tolerance	Tolerance deviation of screen brightness during automatic exposure adjustment.
	Compensation	Target brightness during automatic exposure adjustment.
	EVBias	Exposure deviation value during automatic exposure adjustment, 10bit decimal precision
	AEStrategyMode	Automatic exposure strategy, highlight priority or low light priority.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	AntiflickerEnable	Anti-flash property settings. Anti-flash is disabled by default.
	AntiflickerFrequency	Anti-flicker frequency value
	DebugMode	AE's debug mode parameter
	IRCutOnLv	Environment Lv value without IR Cut, precision is 100
	IRCutOffLv	Use the ambient Lv value of IR Cut with an accuracy of 100
	HistRatioSlope	The blending weight of the brightness of the current frame and the brightness of the highlight area
	MaxHistOffset	After mixing the brightness of the current frame with the brightness of the highlight area, the increased brightness range
	ISONumRangeMin	The minimum value of the ISO num range
	ISONumRangeMax	Maximum value of the ISO num range
	AdjustTargetMax	target max for each LV 50,50, 50, 50, 50, 50, 55, 60, 60, 60, 60, 60, 60, 65, 70, 70
	AdjustTargetMin	target min for each LV 40,40, 40, 40, 40, 40, 45, 50, 50, 50, 50, 50, 50, 55, 60, 60
ExposureInfo	ExposureTime	The current exposure time, in microseconds (us). In FSWDR mode, it indicates the exposure time of the current shortest frame (VS).
	ShortExpTime	In FSWDR mode, it indicates the exposure time of the current short frame (S), in microseconds (us). Linear mode does not care about this value.
	LongExpTime	In FSWDR mode, it indicates the current long frame (L) exposure time in microseconds (us). Linear mode does not care about this value.
	AGain	Current sensor analog gain, 10bit decimal precision.
	DGain	Current sensor digital gain, 10bit decimal precision.
	ISPDGain	Current ISP digital gain, 10bit decimal precision
	AveLum	Average brightness information.
	ISO	Current sensor analog gain*sensor digital gain*ISP digital gain*100, where the accuracy of the gain is 10bit.
	WDRShortAveLuma	WDR Short frame average brightness information.
	WDRExpRatio	WDR long, short frame exposure ratio
	LEFrameAvgLuma	Average brightness of WDR long frame picture

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	SEFrameAvgLuma	Average brightness of WDR short frame picture
	LightValue	Estimated ambient brightness value
AE Route	TotalNum	The number of exposure distribution route nodes, currently the maximum is 16
	AERouteNodex. IntTime	The node exposure time of the node attribute of the exposure distribution route, in microseconds (us)
	AERouteNodex. SysGain	The node gain of the node attribute of the exposure distribution route, including sensor analog gain, sensor digital gain and ISP digital gain, with 10bit precision.
	AERouteNodex. IrisFNO	The number of exposure distribution route nodes, currently the maximum is 16 The node exposure time of the node attribute of the exposure distribution route, in microseconds (us)
	AERouteNodex. IrisFNOLin	The node gain of the node attribute of the exposure distribution route, including sensor analog gain, sensor digital gain and ISP digital gain, with 10bit precision.
WBAttr	ByPass	White balance module Bypass enable, default value CVI_FALSE
	AWBRunInterval	White balance module working frequency
	OpType	Automatic white balance and manual white balance switching
	AlgType	ALG_AWB ALG_AWB_SPEC
	DebugMode	Debug mode
	Rgain	Manual White Balance Red Channel Gain
	Grgain	Manual white balance Gr channel gain
	Gbgain	Manual white balance Gb channel gain
	Bgain	Manual white balance Gb channel gain
	Enable	Auto white balance enabled
	RefColorTemp	The ambient color temperature calibrated by the static white balance coefficient, in Kelvin. It is recommended to obtain 24 color card Raw data for calibration in Macbeth D50 standard light source environment or outdoor sunny environment
	RGStrength	Auto white balance R channel calibration strength
	BGStrength	Auto white balance B channel calibration strength
	Speed	Automatic white balance algorithm convergence speed

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	ZoneSel	When the parameter is | 0 or 255, the white balance algorithm similar to the gray world is used, and other values are used for classification and screening to improve accuracy
	HighColorTemp	Color temperature upper limit of automatic white balance algorithm
	LowColorTemp	Color temperature upper limit of automatic white balance algorithm
	ShiftLimitEn	AWB beyond the white point range of the gain map back to the switch of the white point range
	GainNormEn	Limiting the RGB channel gain can improve the signal-to-noise ratio of low color temperature and low illumination scenes
	NaturalCastEn	AWB style preference switch at low color temperature
	AWBZoneWtEn	Screen partition weight switch, the default is off
	stCTLimit.bEnable	Defines the Gain Range Limit property for self-balancing
	stCTLimit.enOpType	0:AutoMode
	stCTLimit.HighRgLimit	Maximum R gain at high color temperature in manual mode
	stCTLimit.HighBgLimit	Minimum B gain at high color temperature in manual mode
	stCTLimit.LowRgLimit	Minimum R gain at low color temperature in manual mode
	stCTLimit.LowBgLimit	Maximum B gain at low color temperature in manual mode
	stLumaHist.bEnable	AWB brightness and weight parameter switch
	stLumaHist.enOpType	Auto Mode
	stCbCr-Track.bEnable	Linkage parameters of | AWB statistical range and ISO
	Tolerance	The deviation range of automatic white balance adjustment, when the detection error is within the threshold range, AWB will not act
	ZoneRadius	The range of distances used to classify pixels in auto white balance statistics. The smaller the value, the higher the AWB accuracy, but it will reduce the stability of the AWB algorithm
	CurveLLimit	The left limit of the automatic white balance color temperature curve
	CurveRLimit	The right limit of the automatic white balance color temperature curve

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	ExtraLightEn	Whether to consider independent light source points outside the color temperature curve when calculating automatic white balance, up to four independent points
	WhiteBgain	B channel gain of special light source point
	ExpQuant	Judging by the external brightness. ExpQuant is the limit value of the brightness that is turned on. For example ExpQuant = 6, which means that the WB light source point is turned on below LV6 (the general night scene is below LV6) ExpQuant =106 means LV6 and above open ExpQuant =112 means it is enabled above LV12 (LV12 is generally outdoor)
	LightStatus	Types of special light source points, 0: no action 1: Add light source point 2: Delete the calculation near the light source point
	Radius	The area size of the special light source point,
	OutdoorStatus	Indoor or outdoor mode (in manual mode)
	OutThresh	The threshold for judging indoor and outdoor, when the brightness is less than, it is judged as indoor, outdoor LV mostly exceeds 15
	LowStart	Lower the weight of high color temperature, the starting point of high color temperature area is recommended to be 6500K
	LowStop	Lower the weight of the low color temperature, the end point of the low color temperature area is recommended to be 4500K
	HighStart	Lower the weight of high color temperature, the starting point of high color temperature area is recommended to be 6500K
	HighStop	Lower the weight of high color temperature, the end point of high color temperature area is recommended to be 8000K
	GreenEnhanceEn	In the green plant scene, the switch for the green channel is added
	OutShiftLimit	When it is judged as an outdoor scene, the white point range limit of the AWB algorithm
	MultiLightSourceEn	AWB detects whether the current scene is a mixed light source to adjust saturation or CCM
	MultiLSType	Adjust Saturation or CCM
	MultiLSScaler	Adjust saturation or CCM strength when mixing light sources

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	FineTunEn	AWB special color detection switch, such as skin color
	FineTunStrength	Intensity of special color detection such as skin color and blue
	ShiftLimit	When it is judged as an outdoor scene, the white point range limit of the AWB algorithm
	CurvePara	CurvePara[0-2] Planck curve coefficient, given by AWB calibration tool. Planck Lines depict the color performance of a white patch under standard light sources of different color temperatures. CurvePara[3-5] Color temperature curve coefficient, given by AWB calibration tool. color temperature curve The corresponding relationship between the color performance of the painted white block and the color temperature.
	StaticWB	Static white balance coefficient, given by AWB calibration tool.
	AttrZoneWt	32x32 picture weight
	stLumaHist.HistThresh	Threshold for brightness classification (valid in manual mode)
	stLumaHist.HistWt	The weight of brightness classification (valid in manual mode)
	stCbCrTrack.CrMax	The maximum value of R/G under different ISO
	stCbCrTrack.CrMin	The minimum value of R/G under different ISO
	stCbCrTrack.CbMax	The maximum value of B/G under different ISO
	stCbCrTrack.CbMin	The minimum value of B/G under different ISO
	MultiCTBin	Color temperature segmentation parameters
	MultiCTWt	Color Temperature Segment Weight
WBInfo	Rgain	Current R channel gain value
	Ggain	Current G channel gain value
	Bgain	Current B channel gain value
	Saturation	current saturation value
	ColorTemp	Current color temperature value
	InOutStatus	Indoor and outdoor test results
	Bv	Current environment bv value
	CCM	Current color correction matrix value, 8bit decimal precision. Bit 15 is a sign bit, 0 means a positive number, 1 means a negative number, for example, 0x8010 means -16.
BNR	Enable	BNR module enable
	isManualMode	manual mode or automatic mode

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	WindowType	Denoising filtering locality degree. The smaller the value, the more localized the effect.
	DetailSmoothMode	The denoising detail smoothing function is enabled.
	NoiseSuppressStr	Noise suppression strength. The larger the value, the stronger the bright noise removal.
	FilterType	Denoising filter strength. The larger the value, the stronger the bright noise removal.
	NrLscRatio	Adjust the denoising strength with reference to the LSC gain. The larger the value, the more proportional the reference LSC gain is.
	VarThr	Threshold for detecting edges. The larger the value, the fewer the number of edges judged.
	CoringWgtLF	Adjusts the intensity of random noise in the low frequency region. The larger the value, the more noise is preserved in the low frequency region.
	CoringWgtHF	Adjusts the intensity of random noise in the high frequency region. The larger the value, the more noise is preserved in the high-frequency region.
	NonDirFiltStr	Adjusts the strength of noise reduction in the low frequency region. The larger the value, the more noise is removed in the low frequency area.
	VhDirFiltStr	Adjusts the strength of denoising in the horizontal and vertical areas. Larger values remove more noise on horizontal and vertical edges.
	AaDirFiltStr	Adjusts the strength of denoising on diagonal edges. The larger the value, the more noise is removed on the diagonal edges.
Crosstalk	Enable	GE module enable
	isManualMode	manual mode or automatic mode
	Strength	G channel balance global intensity
	FlatThre	Flat area detection node 1-4 threshold
	GrGbDiffThreSec	G channel balance node 1-4 threshold
Demosaic	Enable	Demosaic module enable
	LumaTunedCoringEn	(OpType, CoringEn) (x,0): by noise profile (1,1): by ISO table (0,1): by manual
	isManualMode	manual mode or automatic mode
	CoarseEdgeThr	Coarse edge detection threshold. The smaller the value, the greater the number detected as edges. It is recommended to debug with the parameter CoarseStr.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	CoarseStr	Edge roughness intensity value. The smaller the value, the more directional the processing. Conversely, the more non-directional processing.
	FineEdgeThr	Edge fine-tuning detection threshold. The smaller the value, the greater the number detected as edges. It is recommended to debug with the parameter FineStr.
	FineStr	Edge refinement strength value. The smaller the value, the more directional the processing. Conversely, the more non-directional processing.
	DetailSmoothEnable	Detail smoothing enabled
	DetailSmoothStr	Detail smoothing intensity. The larger the value, the stronger the smoothing strength and the stronger the suppression of false details.
	DetailWgtThr	Detail preservation range threshold. The smaller the value, the greater the range of detail preservation.
	DetailWgtSlope	Details retain intensity. The larger the value, the more detail is preserved.
	DetailWgtMin	Edge detail smoothing preserves the minimum gain allowed.
	DetailWgtMax	Maximum gain allowed for edge detail smoothing preservation.
	LumaWgtThr	Luminance detail smoothing range threshold. The smaller the value, the greater the range of detail smoothing.
	LumaWgtSlope	Brightness Details Smooth Edges Preserve Strength. The larger the value, the stronger the detail smoothing.
	LumaWgtMin	Minimum gain allowed for luminance detail smoothing strength.
	LumaWgtMax	Maximum gain allowed for luminance detail smoothing strength.
	EdgeEnhanceEnable	Edge hardening enabled
	OverShtGain	Adjust the degree of overshoot. The larger the value, the greater the edge enhancement. When OverShtGain is doubled, its value is 256.
	UnderShtGain	Adjust the degree of undershoot. The larger the value, the greater the edge enhancement. When UnderShtGain is doubled, its value is 256.

continues on next page

Table 44.1 – continued from previous page

Parameter		Description
	NoiseSuppressStr	Noise suppression strength. The larger the value, the more noise can be suppressed from being strengthened.
	GainTable	Non-directional midrange texture boosts intensity. The higher the value, the sharper the non-directional mid-frequency texture. This parameter is an array | of 33, represented as a continuous 33-segment intensity curve. When the value is 128, the gain is doubled.
	LumaTunedCoring	Noise suppression strength. The larger the value, the more noise can be suppressed from being strengthened.
AE config	W i n W e i g h t	AE metering weight

44.3.2 LEVEL2 Debugging Information Analysis

On the basis of level1, we add some additional information about meshLscGainLut, exposureInfo, and GammaTable

[parameter Description]

Parameter		Description
Module/GammaEx	GammaTable	Node value of Gamma curve
ShadingEx	MeshLscGainLut	LSC grid form red channel compensation gain table, including R, G, B three channels
ExposureInfoEx	Exposure	Current exposure. It is equal to the product of exposure time and exposure gain, where the unit of exposure time is the number of exposure lines, and the exposure gain is 6bit decimal precision.
	ExposureIsMax	0: ISP does not reach the maximum exposure level; 1: ISP reaches the maximum exposure level.
	HistError	Statistical information, the difference between the target brightness value and the actual value of AE. A positive value indicates that the current expected brightness information is greater than the actual brightness information, and a negative value indicates that the expected brightness information is less than the actual brightness information.
	PirisFno	The equivalent gain corresponding to the current F value of P-Iris aperture.
	Fps	Actual image frame rate * 100。
	RefExpRatio	The reference exposure ratio which is used to estimate the dynamic range of the current scene.

44.3.3 LEVEL3 debugging information analysis

Based on Level 2, the statistical value of 3a is added.

[Parameters Description]

Parameter		Description
AE statistics	LEGlobalAvgR	Statistical average of R component in global statistics of long exposure frames
	LEGlobalAvgGr	Statistical mean of GR components in global statistics of long exposure frames
	LEGlobalAvgGb	Statistical average of GB components in global statistics of long exposure frames
	LEGlobalAvgB	Statistical mean value of B component in global statistics of long exposure frame
	SEGlobalAvgR	Statistical average of R component in global statistics of short exposure frames
	SEGlobalAvgGr	Statistical mean of GR components in global statistics of short exposure frames
	SEGlobalAvgGb	Statistical average of GB components in global statistics of short exposure frames
	SEGlobalAvgB	Statistical mean value of B component in global statistics of short exposure frame
	LEHistogramMem	Pixel histogram statistics of global statistics for long exposure frames

continues on next page

Table 44.2 – continued from previous page

Parameter		Description
	SEHistogramMem	Pixel histogram statistics of global statistics for short exposure frames
	LEZoneRAvg	Statistical average of R-channel components in long exposure frame
	LEZoneGrAvg	Statistical average of Gr channel components in long exposure frame
	LEZoneGbAvg	Statistical average of Gb channel components in long exposure frame
	LEZoneBAvg	Statistical average of B-channel components in long exposure frame
	SEZoneRAvg	Statistical average of R-channel components in short exposure frame
	SEZoneGrAvg	Statistical average of Gr channel components in short exposure frame
	SEZoneGbAvg	Statistical average of Gb channel components in short exposure frame
	SEZoneBAvg	Statistical average of B-channel components in short exposure frame
AWB statistics	LEGlobalR	R-component statistical mean of global statistics for long exposure frames
	LEGlobalG	G-component statistical mean of global statistics for long exposure frames

continues on next page

Table 44.2 – continued from previous page

Parameter		Description
	LEGlobalB	B-component statistical mean of global statistics for long exposure frames
	LECountAll	Pixel statistical average of global statistics for long exposure frames
	SEGlobalR	R-component statistical mean of global statistics for short exposure frames
	SEGlobalG	G-component statistical mean of global statistics for short exposure frames
	SEGlobalB	B-component statistical mean of global statistics for short exposure frames
	SECountAll	Pixel statistical average of global statistics for short exposure frames
	LEZoneAvgR	The statistical average of R component in long exposure frame
	LEZoneAvgG	The statistical average of G component in long exposure frame
	LEZoneAvgB	The statistical average of B component in long exposure frame
	LEZoneCountAll	Pixel statistics of long exposure frame
	SEZoneAvgR	The statistical average of R component in short exposure frame
	SEZoneAvgG	The statistical average of G component in short exposure frame

continues on next page

Table 44.2 – continued from previous page

Parameter		Description
	SEZoneAvgB	The statistical average of B component in short exposure frame
	SEZoneCountAll	Pixel statistics of short exposure frame

45 Overview of 3A Development Guide

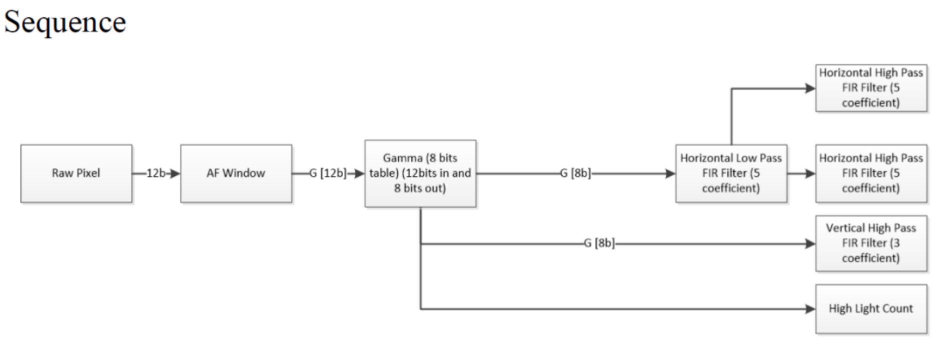
46 3A development User Guide

46.1 AF Statistics Instructions

46.1.1 Function Overview

Auto focus is to get the current image clarity value FV(Focus Value) by analyzing the characteristics of the image. Because the clearer the image is, the greater the FV is. Therefore, by comparing the clarity value of each position, the highest point of FV value curve can be found, which is the focus. After finding the focus, the user can control the focusing horse to reach the best position to complete auto focus.

At present, AF provides four filters and brightness information, which are the horizontal direction H1, H2 and the vertical direction v1. In the horizontal direction, a low-pass filter is used first, and then two high pass filters are used to get the values of H1 and H2. In the vertical direction, only high pass filters are used. Currently, AF only supports statistical information in Bayer domain.



46.1.2 Crop Input Image

AF supports clipping of input image. Users can determine the current statistical area of AF by X, y, W and h in stCrop. For details, refer to ISP_STATISTICS_CFG_S for relevant structure information

46.1.3 Configuration of Bayer Domain

There are two pre-processing methods for data before entering AF module

1. Users can choose whether to go through gamma or not. If necessary, they need to enable gamma and fill in the appropriate gamma curve
2. User can choose whether to turn on pre filtering to eliminate the influence of salt and pepper noise on statistics

46.1.4 Suppress the Influence of Light Source on FV Value

When there is a point light source in the image, the FV value will be affected by the halo diffusion, but the FV value will increase.

In order to suppress this phenomenon, u16HlCnt is added to count the value of high brightness points in the window.

The user can adjust u16HighLumaTh to determine the threshold value of highlight points.

When the window is blurred, the number of highlights in the window increases, and when the window is clear, the number of highlights in the window is the least.

The user can use this information to determine the most suitable focus

46.1.5 Notes on Statistical Information Configuration

Type	Description
Block size	Maxiumum 17 * 15
Work domain of statistics module	RAW
Whether Bayer statistical parameter subtracts black level	Subtracted

46.1.6 Acquisition of FV Value

When the last pixel of the image passes, the statistical information can be obtained, and the user can synchronously obtain the statistical value through CVI_ISP_GetVDTimeOut.

Please refer to the process of 33.1.8.

46.1.7 Calculation of FV Value

There are three kinds of statistical values that can be obtained by a block: H_0 obtained by the first group of Horizontal filters, H_1 obtained by the second group of Horizontal filters, and V_0 obtained by the Vertical filters. We name the FV value of each block FV_n , and set their own weights for each statistical value, which are W0 / W1 / W2 respectively. Then the FV_n value is

$$FV_n = \frac{W_0 * H0_n + W_1 * H1_n + W_2 * V0_n}{W_0 + W_1 + W_2}$$

The final FV value also needs to add weight to the FV of each block. Assuming that the weight of the nth block is W_n, the final FV value is:

$$FV = \frac{\sum_{n=0}^{blocks} FV_n * W_n}{\sum_{n=0}^{blocks} W_n}$$

46.1.8 FV Calculation Reference Code

```
ISP_AF_STATISTICS_S afStat;
CVI_U32 row, col;
CVI_S32 s32Ret = CVI_SUCCESS;
CVI_U64 stsValue = 0;
VI_PIPE ViPipe = 0;
ISP_VD_TYPE_E enIspVDType = ISP_VD_FE_START;
CVI_CHAR input[10];
ISP_STATISTICS_CFG_S stsCfg;
ISP_PUB_ATTR_S stPubAttr;
struct timeval t1, t2;
// AF weighting table
static int AFWeight[15][17] = {{1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                               {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
```

(continues on next page)

(continued from previous page)

```
                              {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                              {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                              {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                              {1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,1},
                              {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}};
// Get current statistic and related size setting.
s32Ret = CVI_ISP_GetStatisticsConfig(ViPipe, &stsCfg);
s32Ret |= CVI_ISP_GetPubAttr(ViPipe, &stPubAttr);
if (s32Ret != CVI_SUCCESS) {
        CVI_TRACE_LOG(CVI_DBG_ERR, "Get Statistic info fail with %#x!\n",␣
↪s32Ret);
        return s32Ret;
}
// Config AF Enable.
stsCfg.stFocusCfg.stConfig.bEnable = 1;
// Config low pass filter.
stsCfg.stFocusCfg.stConfig.u8HFltShift = 0;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[0] = 0;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[1] = 1;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[2] = 2;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[3] = 3;
stsCfg.stFocusCfg.stConfig.s8HVFltLpCoeff[4] = 4;
// Config gamma enable.
stsCfg.stFocusCfg.stConfig.stRawCfg.PreGammaEn = 0;
// Config pre NR enable.
stsCfg.stFocusCfg.stConfig.stPreFltCfg.PreFltEn = 1;
// Config H & V window.
stsCfg.stFocusCfg.stConfig.u16Hwnd = 17;
stsCfg.stFocusCfg.stConfig.u16Vwnd = 15;
// Config crop related setting. Has some limitation
stsCfg.stFocusCfg.stConfig.stCrop.bEnable = 1;
stsCfg.stFocusCfg.stConfig.stCrop.u16X = 8;
stsCfg.stFocusCfg.stConfig.stCrop.u16Y = 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16W = stPubAttr.stWndRect.u32Width - 8 * 2;
stsCfg.stFocusCfg.stConfig.stCrop.u16H = stPubAttr.stWndRect.u32Height - 2 * 2;
// Config first horizontal high pass filter.
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[1] = -3;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[2] = 0;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[3] = -10;
stsCfg.stFocusCfg.stHParam_FIR0.s8HFltHpCoeff[4] = 0;
// Config 2nd horizontal high pass filter.
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[0] = 0;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[1] = -3;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[2] = 0;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[3] = -10;
stsCfg.stFocusCfg.stHParam_FIR1.s8HFltHpCoeff[4] = 0;
// Config vertical high pass filter.
```

(continues on next page)

(continued from previous page)

```
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[0] = 8;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[1] = -15;
stsCfg.stFocusCfg.stVParam_FIR.s8VFltHpCoeff[2 ] = 0;
stsCfg.unKey.bit1FEAfStat = 1;
s32Ret = CVI_ISP_SetStatisticsConfig(ViPipe, &stsCfg);
if (s32Ret != CVI_SUCCESS) {
        CVI_TRACE_LOG(CVI_DBG_ERR, "ISP Set Statistic failed with %#x!\n",␣
↪s32Ret);
        return s32Ret;
}

printf("select. c -> Fv curve\n");
printf("........ h0 -> print h0 blocks statistic\n");
printf("........ h1 -> print h1 blocks statistic\n");
printf("........ v0 -> print v0 blocks statistic\n");
printf("........ hlc -> print hlcnt blocks statistic\n");
scanf("%s", input);

while(1) {
        // Wait VD start for get focus statistic data.
        s32Ret = CVI_ISP_GetVDTimeOut(ViPipe, enIspVDType, 5000);
        s32Ret |= CVI_ISP_GetFocusStatistics(ViPipe, &afStat);
        if (s32Ret != CVI_SUCCESS) {
                CVI_TRACE_LOG(CVI_DBG_ERR, "Get Statistic failed with %#x\n",␣
↪s32Ret);
                return CVI_FAILURE;
        }
        // print each focus statistic.
        if (strncmp(input, "c", 1) != 0) {
                for (row = 0; row < AF_ZONE_ROW; row++) {
                        for (col = 0; col < AF_ZONE_COLUMN; col++) {
                                if (strncmp(input, "h0", 2) == 0) {
                                        stsValue = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u64h0;
                                } else if (strncmp(input, "h1", 2) == 0) {
                                        stsValue = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u64h1;
                                } else if (strncmp(input, "v0", 2) == 0) {
                                        stsValue = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u32v0;
                                } else {
                                        stsValue = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u16HlCnt;
                                }
                                printf("%d ", stsValue);
                        }
                        printf("\n");
                }
```

(continues on next page)

(continued from previous page)

```
                  continue;
         }

         CVI_U64 FVn = 0, FV = 0;
         CVI_U32 totalWeightSum = 0;
         // weight for each statistic
         const CVI_U32 weight1 = 1, weight2 = 1, weight3 = 1;
         const CVI_U32 blockWeightSum = weight1 + weight2 + weight3;
         // calculate AF statistics
         for (row = 0; row < AF_ZONE_ROW; row++) {
                  for (col = 0; col < AF_ZONE_COLUMN; col++) {
                           CVI_U64 h0 = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u64h0;
                           CVI_U64 h1 = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u64h1;
                           CVI_U32 v0 = afStat.stFEAFStat.
↪stZoneMetrics[row][col].u32v0;

                           FVn = (weight1 * h0 + weight2 * h1 + weight3 * v0) /␣
↪blockWeightSum;
                           FV += FVn * AFWeight[row][col];
                           totalWeightSum += AFWeight[row][col];
                  }
         }
         FV = FV / totalWeightSum;
}
return CVI_SUCCESS;
```

47 Developer' s Guide

47.1 API Reference

The statistical information can be obtained after the last pixel of the image passes, and the user can use `CVI_ISP_GetVDTimeOut` to get statistics synchronously.

Refer to the process of 33.1.8. Because the user space task scheduling of Linux system can not guarantee the consistent real-time performance, the driver configuration needs to be completed in kernel space.

ISP provides the registration of synchronous callback interface to synchronize with VD.

In this chapter, there are corresponding interface descriptions.

Users can put tasks with high real-time requirements in synchronous callbacks. The bottom layer provides HwIRQ and Workqueue to implement them. You can choose the corresponding implementation method to determine the real-time level.

HwIRQ means that tasks are implemented in interrupt service, which has the highest real-time performance. The real-time performance of Workqueue depends on Linux system call.

- *isp_sync_task_register* : Register synchronization callback interface with ISP.
- *isp_sync_task_unregister* : Unregister the synchronization callback interface with ISP.

47.1.1 isp_sync_task_register

【Description】

Register synchronization callback interface with ISP.

【Syntax】

```
int isp_sync_task_register(int vi_pipe, struct isp_sync_task_node *new_node);
```

【Parameter】

Parameter	Description	Input/Output
vi_pipe	VI_PIPE number	Input
new_node	Newly inserted synchronization callback node	Input

【Retrun Value】

Return Value	Description
0	Success
non 0	Failure.

【Requirement】

- Header files: cvi_vip_isp_ext.h

【Note】

Make sure ISP driver is loaded before use.

Because the internal implementation of ISP synchronization callback does not save the user’s incoming new_Node refers to the entity, so isp_sync_task_node is required to be used to define the entity, which cannot be a local variable.

【Example】

None.

【Related Topic】

- *isp_sync_task_unregister*

47.1.2 isp_sync_task_unregister

【Description】

Unregister the synchronization callback interface with ISP.

【Syntax】

```
int isp_sync_task_unregister(int vi_pipe, struct isp_sync_task_node *del_node);
```

【Parameter】

Parameter	Description	Input/Output
vi_pipe	VI_PIPE number	Input
del_node	Synchronization callback node to be deleted	Input

【Retrun Value】

Return Value	Description
0	Success
non 0	Failure.

【Requirement】

- Header files: cvi_vip_isp_ext.h

【Note】

Make sure ISP driver is loaded before use.

【Example】

None.

【Related Topic】

- *isp_sync_task_register*

47.2 Data Types

- *isp_sync_tsk_method* :Define synchronous callback method to determine real-time performance.
- *isp_sync_task_node* :Define the synchronization callback node information.

47.2.1 isp_sync_tsk_method

【Description】

Define synchronous callback method to determine real-time performance.

【Syntax】

```
enum isp_sync_tsk_method {
    ISP_SYNC_TSK_METHOD_HW_IRQ = 0,
    ISP_SYNC_TSK_METHOD_WORKQUE,
    ISP_SYNC_TSK_METHOD_BUTT
};
```

【Member】

Member	Description
ISP_SYNC_TSK_METHOD_HW_IRQ	Use hardware interrupt to call back.
ISP_SYNC_TSK_METHOD_HW_WORKQUE	Use work queue callback.

【Note】

None.

【Related Data Type and Interface】

None.

47.2.2 isp_sync_task_node

【Description】

Define the synchronization callback node information.

【Syntax】

```
struct isp_sync_task_node {
    enum isp_sync_tsk_method method;
    __s32 (*isp_sync_tsk_call_back)(__u64 data);
    __u64 data;
    const char *sz_id;
    struct list_head list;
};
```

【Member】

Member	Description
method	Callback mode.
isp_sync_tsk_call_back	Callback function, which is passed in when the user registers.
data	Callback function parameter, which is passed in when the user registers.
sz_id	Node ID
list	List node, used to manage multiple callback nodes, no need to pay attention.

【Example】

```
isp_sync_task_node sync_node = {
    .method = ISP_SYNC_TSK_METHOD_HW_IRQ,
    .isp_sync_tsk_call_back = sync_af_calc,
    .data = 0,
    .sz_id = "hw_0"
};
```

【Note】

None.

【Related Data Type and Interface】

- *isp_sync_tsk_method*

48 Appendix

49 Abbreviations