



# CV186AH IVE API 使用手册

Version: 1.1

Release date: 2023/12

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

# 目录

<b>1</b>	<b>声明</b>	<b>2</b>
<b>2</b>	<b>功能概述</b>	<b>3</b>
2.1	目的 . . . . .	3
2.2	定义及缩写 . . . . .	3
<b>3</b>	<b>API 参考</b>	<b>10</b>
3.1	Create Handle . . . . .	10
3.2	Destroy Handle . . . . .	10
3.3	DMA . . . . .	11
3.4	Filter . . . . .	14
3.5	Filter And CSC . . . . .	15
3.6	CSC . . . . .	16
3.7	Sobel . . . . .	17
3.8	NormGrad . . . . .	19
3.9	Canny Edge . . . . .	20
3.10	Canny Hysteresis Edge . . . . .	21
3.11	MagAndAng . . . . .	22
3.12	Dilate . . . . .	23
3.13	Erode . . . . .	24
3.14	Thresh . . . . .	25
3.15	And . . . . .	26
3.16	Sub . . . . .	27
3.17	Or . . . . .	28
3.18	Map . . . . .	29
3.19	OrdStatFilter . . . . .	30
3.20	Integral . . . . .	31
3.21	Histogram . . . . .	32
3.22	Add . . . . .	33
3.23	Xor . . . . .	34
3.24	Match BgModel . . . . .	35
3.25	Update BgModel . . . . .	36
3.26	Gradient of Foreground . . . . .	37
3.27	GMM . . . . .	38
3.28	GMM2 . . . . .	39
3.29	Bernsen . . . . .	41
3.30	NCC . . . . .	42

3.31	CCL	43
3.32	LBP	44
3.33	SAD	45
3.34	BufFlush	47
3.35	BufRequest	47
3.36	CreateMemInfo	48
3.37	CreateDataInfo	49
3.38	CreateImage	49
3.39	CreateImage with Cache	50
3.40	ResetImage	51
3.41	ReadImageArray	51
3.42	ReadMem	52
3.43	ReadMemArray	53
3.44	ReadData	54
3.45	ReadDataArray	54
3.46	ReadImage	55
3.47	ReadRawImage	56
3.48	WriteData	56
3.49	WriteMem	57
3.50	WriteImage	58
3.51	WriteRawImage	58
3.52	Reset Register	59
3.53	Dump Register	60
3.54	Split DiffFg of BgModel	60
3.55	Split ChgSta of BgModel	61
3.56	Query Tasks	62
3.57	Image2VideoFrameInfo	62
3.58	VideoFrameInfo2Image	63
3.59	FreeM	64
3.60	FreeI	64
3.61	FreeD	65
3.62	Thresh_S16	66
3.63	Thresh_U16	67
3.64	Resize	68
3.65	16BitTo8Bit	69
3.66	RGB YUV Erode to Dilate	70
3.67	STCandiCorner	71
3.68	Background Subtraction	71
3.69	Compare Ive Image	72
3.70	Compare Ive Mem	73
3.71	Compare Ive Data	74
3.72	Compare Ive SAD Image	74
<b>4</b>	<b>数据类型和数据结构</b>	<b>76</b>
4.1	定义数据类型	78
4.2	定义结构类型	79
4.2.1	IVE_IMAGE_TYPE_E	79

4.2.2	IVE_IMAGE_S . . . . .	81
4.2.3	IVE_SRC_IMAGE_S . . . . .	82
4.2.4	IVE_DST_IMAGE_S . . . . .	82
4.2.5	IVE_DATA_S . . . . .	83
4.2.6	IVE_SRC_DATA_S . . . . .	84
4.2.7	IVE_DST_DATA_S . . . . .	84
4.2.8	IVE_MEM_INFO_S . . . . .	85
4.2.9	IVE_SRC_MEM_INFO_S . . . . .	85
4.2.10	IVE_DST_MEM_INFO_S . . . . .	86
4.2.11	IVE_8BIT_U . . . . .	86
4.2.12	IVE_POINT_U16_S . . . . .	87
4.2.13	IVE_POINT_S16_S . . . . .	88
4.2.14	IVE_RECT_U16_S . . . . .	88
4.2.15	IVE_LOOK_UP_TABLE_S . . . . .	89
4.2.16	IVE_DMA_MODE_E . . . . .	90
4.2.17	IVE_DMA_CTRL_S . . . . .	91
4.2.18	IVE_FILTER_CTRL_S . . . . .	92
4.2.19	IVE_CSC_MODE_E . . . . .	93
4.2.20	IVE_CSC_CTRL_S . . . . .	94
4.2.21	IVE_SOBEL_OUT_CTRL_E . . . . .	95
4.2.22	IVE_SOBEL_CTRL_S . . . . .	95
4.2.23	IVE_MAG_AND_ANG_OUT_CTRL_E . . . . .	96
4.2.24	IVE_MAG_AND_ANG_CTRL_S . . . . .	97
4.2.25	IVE_DILATE_CTRL_S . . . . .	97
4.2.26	IVE_ERODE_CTRL_S . . . . .	98
4.2.27	IVE_THRESH_MODE_E . . . . .	98
4.2.28	IVE_THRESH_CTRL_S . . . . .	100
4.2.29	IVE_SUB_MODE_E . . . . .	101
4.2.30	IVE_SUB_CTRL_S . . . . .	101
4.2.31	IVE_INTEG_OUT_CTRL_E . . . . .	102
4.2.32	IVE_INTEG_CTRL_S . . . . .	103
4.2.33	IVE_THRESH_S16_MODE_E . . . . .	103
4.2.34	IVE_THRESH_S16_CTRL_S . . . . .	105
4.2.35	IVE_THRESH_U16_MODE_E . . . . .	106
4.2.36	IVE_THRESH_U16_CTRL_S . . . . .	107
4.2.37	IVE_16BIT_TO_8BIT_MODE_E . . . . .	108
4.2.38	IVE_16BIT_TO_8BIT_CTRL_S . . . . .	109
4.2.39	IVE_ORD_STAT_FILTER_MODE_E . . . . .	109
4.2.40	IVE_ORD_STAT_FILTER_CTRL_S . . . . .	110
4.2.41	IVE_EQUALIZE_HIST_CTRL_S . . . . .	111
4.2.42	IVE_MAP_MODE_E . . . . .	111
4.2.43	IVE_MAP_CTRL_S . . . . .	112
4.2.44	IVE_MAP_U8BIT_LUT_MEM_S . . . . .	113
4.2.45	IVE_MAP_U16BIT_LUT_MEM_S . . . . .	113
4.2.46	IVE_MAP_S16BIT_LUT_MEM_S . . . . .	114
4.2.47	IVE_EQUALIZE_HIST_CTRL_MEM_S . . . . .	114
4.2.48	IVE_CANNY_HYS_EDGE_CTRL_S . . . . .	115

4.2.49	IVE_CANNY_STACK_SIZE_S	116
4.2.50	IVE_ADD_CTRL_S	116
4.2.51	IVE_NCC_DST_MEM_S	117
4.2.52	IVE_REGION_S	118
4.2.53	IVE_CCBLOB_S	119
4.2.54	IVE_CCL_MODE_E	120
4.2.55	IVE_CCL_CTRL_S	120
4.2.56	IVE_GMM_CTRL_S	121
4.2.57	IVE_LBP_CMP_MODE_E	122
4.2.58	IVE_LBP_CTRL_S	123
4.2.59	IVE_NORM_GRAD_OUT_CTRL_E	124
4.2.60	IVE_NORM_GRAD_CTRL_S	125
4.2.61	IVE_ST_CANDI_CORNER_CTRL_S	125
4.2.62	IVE_ST_CORNER_INFO_S	126
4.2.63	IVE_ST_CORNER_CTRL_S	127
4.2.64	IVE_GRAD_FG_MODE_E	127
4.2.65	IVE_FG_STAT_DATA_S	128
4.2.66	IVE_BG_STAT_DATA_S	129
4.2.67	IVE_SAD_MODE_E	129
4.2.68	IVE_SAD_OUT_CTRL_E	130
4.2.69	IVE_SAD_CTRL_S	131
4.2.70	IVE_RESIZE_MODE_E	132
4.2.71	IVE_RESIZE_CTRL_S	133
4.2.72	IVE_HOG_CTRL_S	134
4.2.73	IVE_16BIT_TO_8BIT_MODE_E	134
4.2.74	IVE_16BIT_TO_8BIT_CTRL_S	135
4.2.75	IVE_BLOCK_CTRL_S	136
<b>5</b>	<b>技巧说明</b>	<b>138</b>
5.1	额外的缓冲区	138
<b>6</b>	<b>FAQ</b>	<b>139</b>
6.1	Cache 内存的使用	139
6.2	bInstant 参数的设定	139

## 修订记录

Version	Date	Description
1.0	2022/06/18	初版
1.1	2023/12/19	增加 ccl 功能，修正更新

# 1 声明

---



## 法律声明

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

本文件内信息如有更改，恕不另行通知。晶视智能不对使用或依赖本文件所含信息承担任何责任。

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

## 联系我们

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

深圳市宝安区福海街道展城社区会展湾云岸广场 T10 栋

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

**邮编** 100094（北京）518100（深圳）

**官方网站** <https://www.sophgo.com/>

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

# 2 功能概述

---

## 2.1 目的

智能视频引擎 Intelligent Video Engine (IVE) 是一种使用硬件去加速电脑视觉算法的模块，用户利用 IVE 开发智能分析方案可以加速智能分析的运算，降低处理器占用。当前 IVE 所提供的算子可以支撑开发影像或视频的智能分析方案。

## 2.2 定义及缩写

- 句柄 (handle)

用户在调用算子创建任务时，系统会为每个任务分配一个 handle，用于标识不同的任务的执行状态。

- 返回结果标志 (bInstant)

True 表示 Busy waiting mode, False 表示 Interrupt mode。

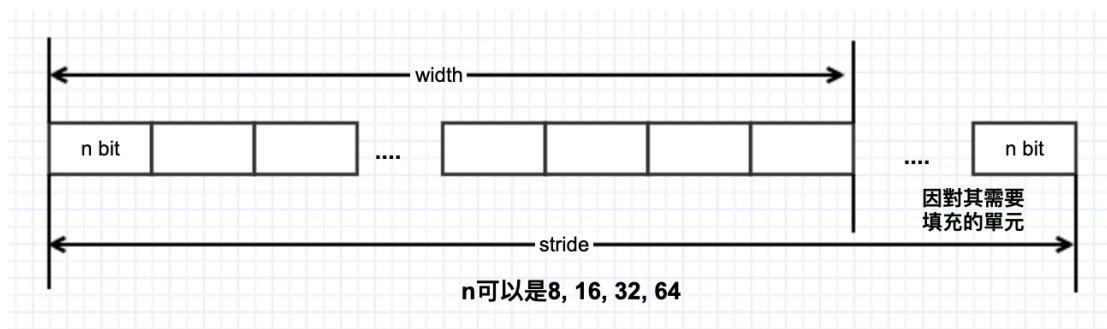
- 跨度 (stride)

与图像或二维数据的 width 度量一致的量，如图 1-1 所示。

- IVE\_IMAGE\_S 图像数据跨度，表示图像一行以“像素”计算的单元个数，“像素”位宽可以是 8bit, 16bit 等。
- IVE\_DATA\_S 二维数据跨度，表示二维数据一行的字节数，即为图 1-1 中  $n=8$  的情况。

图 1-1 跨度 (stride) 示意图





### · 对齐

硬件为了快速访问内存首地址或者跨行访问数据，要求内存地址或内存跨度必须为对齐系数的倍数。

#### – 数据内存首地址对齐

当前 IVE 算子对其输入输出要求 16 像素对齐

#### – 跨度对齐

对于二维广义图像、二维单分量数据以及一维数组数据的跨度均必须满足 16 像素对齐。

输入输出数据类型

类型	图像描述	内存地址	跨度
IVE_IMAGE_TYPE_8UC1	8UC1 无符号单通道图像 图 1-2	仅用到 IVE_IMAGE_S 中 u64PhyAddr[0]、u64VirAddr[0]	仅用到 u32Stride[0]
IVE_IMAGE_TYPE_8SC1	8SC1 有符号单通道图像 图 1-2	仅用到 IVE_IMAGE_S 中 u64PhyAddr[0]、u64VirAddr[0]	仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUVC420SP	YUVC420SP Semi-Planar 数据格式图像	内存地址用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0](亮度 Y), u64PhyAddr[1]、u64VirAddr[1](色度 U, V)	跨度用到 u32Stride[0](亮度跨度)、u32Stride[1](色度 U, V 跨度)
IVE_IMAGE_TYPE_YUVC422SP	YUVC422SP Semi-Planar 数据格式图像	内存地址用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0](亮度 Y), u64PhyAddr[1]、u64VirAddr[1](色度 U, V)	跨度用到 u32Stride[0](亮度跨度)、u32Stride[1](色度 U, V 跨度)

下页继续

表 2.1 – 续上页

类型	图像描述	内存地址	跨度
IVE_IMAGE_TYPE_YUV420P	YUV420P Planar 数据格式图像, 图 1-3	内存地址用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0](亮度 Y), u64PhyAddr[1]、u64VirAddr[1](色度 U) 和 u64PhyAddr[2]、u64VirAddr[2](色度 V)	跨度用到 u32Stride[0](亮度跨度)、u32Stride[1](色度 U 跨度) 和 u32Stride[2](色度 V 跨度)
IVE_IMAGE_TYPE_YUV422P	YUV422P Planar 数据格式图像, 图 1-4	内存地址用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0](亮度 Y), u64PhyAddr[1]、u64VirAddr[1](色度 U) 和 u64PhyAddr[2]、u64VirAddr[2](色度 V)	跨度用到 u32Stride[0](亮度跨度)、u32Stride[1](色度 U 跨度) 和 u32Stride[2](色度 V 跨度)
IVE_IMAGE_TYPE_YUV444P	YUV444P Planar 数据格式图像, 图 1-5	内存地址用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0](亮度 Y), u64PhyAddr[1]、u64VirAddr[1](色度 U) 和 u64PhyAddr[2]、u64VirAddr[2](色度 V)	跨度用到 u32Stride[0](亮度跨度)、u32Stride[1](色度 U 跨度) 和 u32Stride[2](色度 V 跨度)
IVE_IMAGE_TYPE_YUV420P_PACKAGE	YUV420P Planar 数据格式图像, 图 1-3	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV422P_PACKAGE	YUV422P Planar 数据格式图像, 图 1-4	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV444P_PACKAGE	YUV444P Planar 数据格式图像, 图 1-5	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV420P_PLANAR	YUV420P Planar 数据格式图像, 图 1-3	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV422P_PLANAR	YUV422P Planar 数据格式图像, 图 1-4	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV444P_PLANAR	YUV444P Planar 数据格式图像, 图 1-5	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV420P_PACKAGE	YUV420P Planar 数据格式图像, 图 1-3	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV422P_PACKAGE	YUV422P Planar 数据格式图像, 图 1-4	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]
IVE_IMAGE_TYPE_YUV444P_PACKAGE	YUV444P Planar 数据格式图像, 图 1-5	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0]	跨度仅用到 u32Stride[0]

下页继续



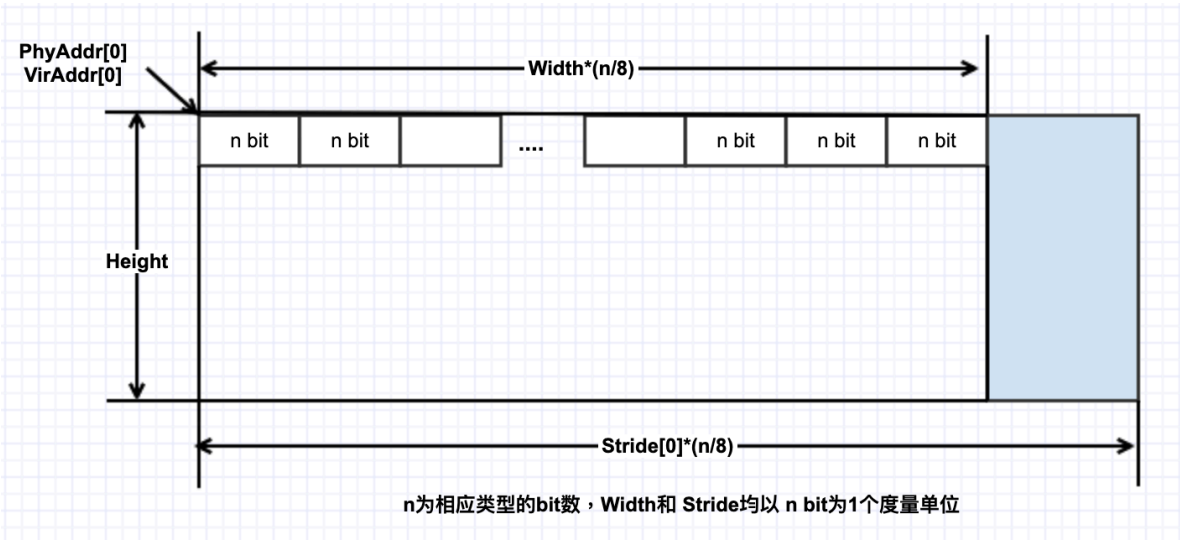


图 1-3 IVE\_IMAGE\_TYPE\_YUV420P 类型的 IVE\_IMAGE\_S 图像

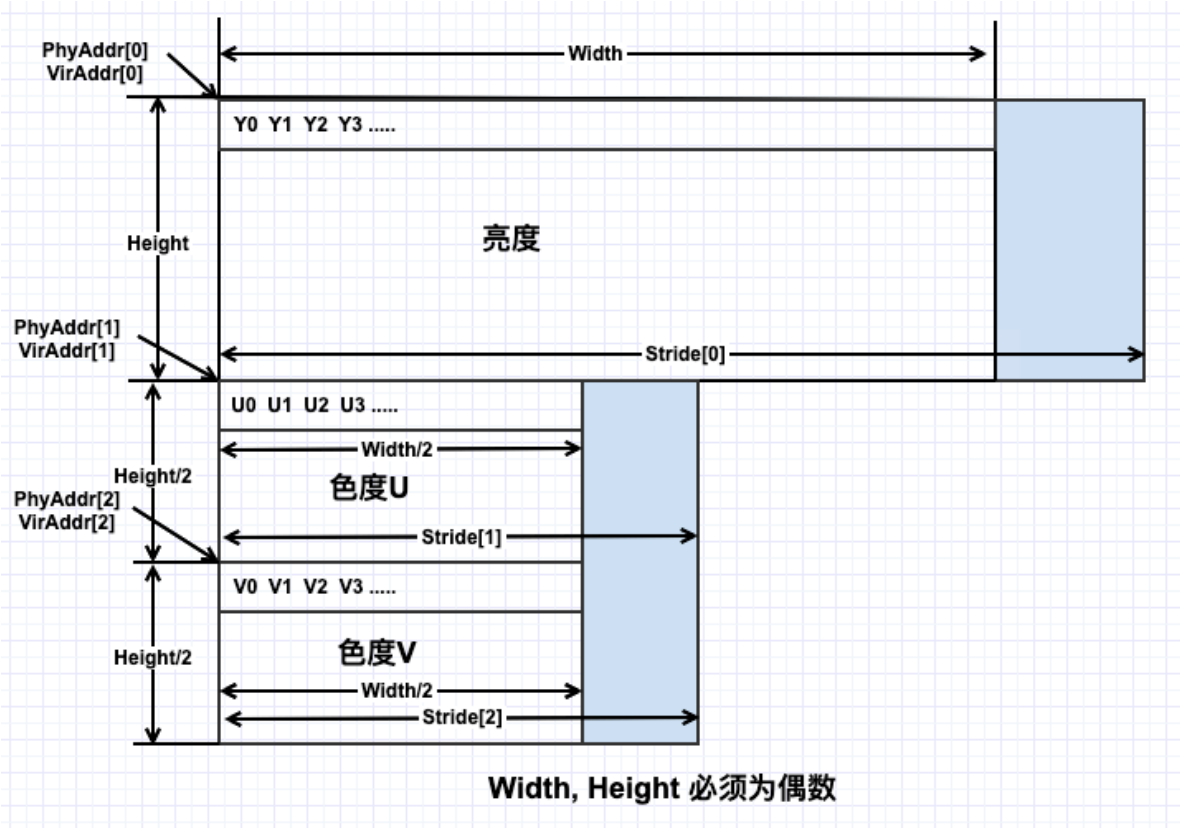


图 1-4 IVE\_IMAGE\_TYPE\_YUV422P 类型的 IVE\_IMAGE\_S 图像

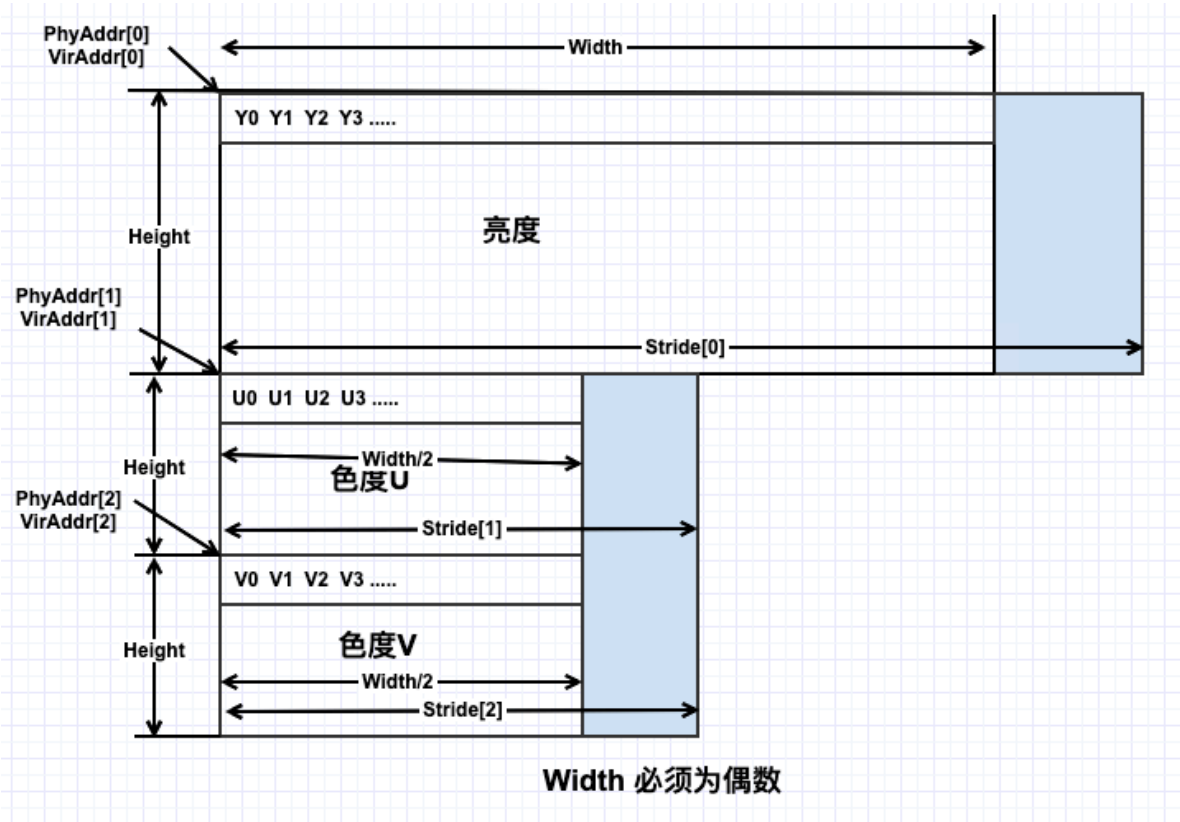


图 1-5 IVE\_IMAGE\_TYPE\_U8C3\_PACKAGE 类型的 IVE\_IMAGE\_S 图像

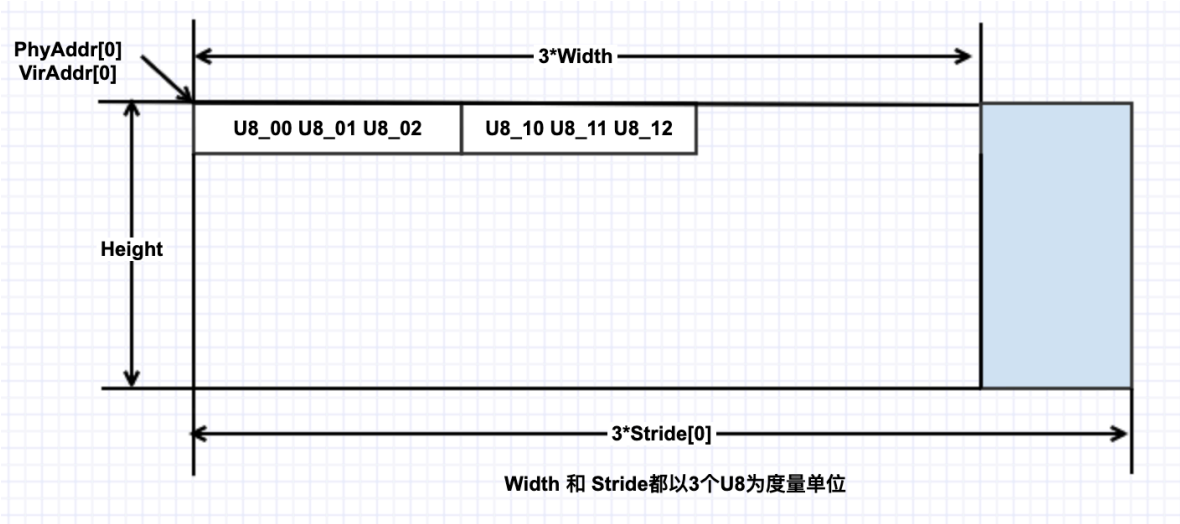
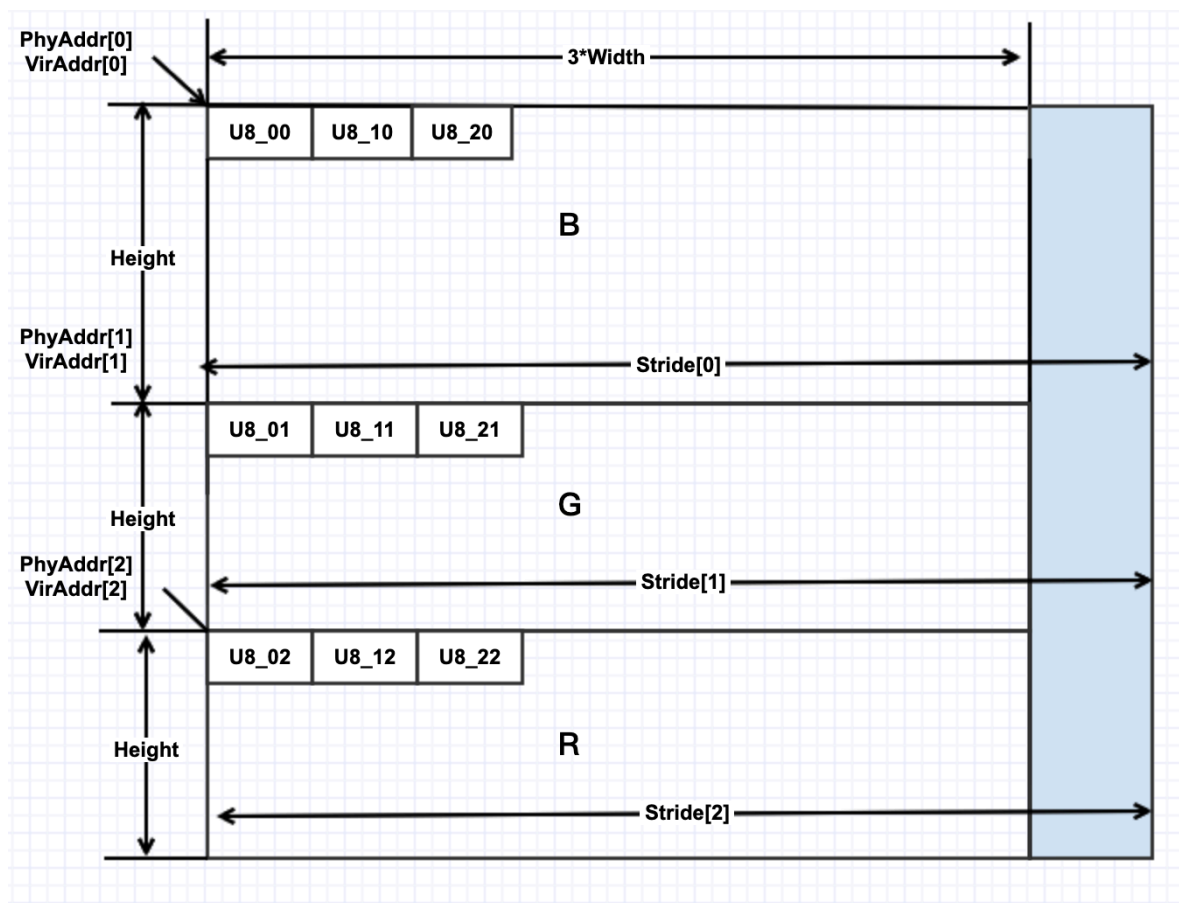


图 1-6 IVE\_IMAGE\_TYPE\_U8C3\_PLANAR 类型的 IVE\_IMAGE\_SRC 图像



# 3 API 参考

## 3.1 Create Handle

**【描述】**

创建 IVE 句柄。

**【语法】**

```
IVE_HANDLE CVI_IVE_CreateHandle();
```

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.2 Destroy Handle

**【描述】**

释放 IVE 句柄。

**【语法】**

```
CVI_S32 CVI_IVE_DestroyHandle(IVE_HANDLE pIveHandle);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 【返回值】

返回值	描述
0	成功
非 0	失败

## 3.3 DMA

## 【描述】

创建直接内存访问任务，支持快速拷贝、间隔拷贝、内存填充: 可实现数据从一块内存快速拷贝到另一块内存，或者从一块内存有规律的拷贝一些数据到另一块内存，或者对一块内存进行填充操作。

## 【语法】

```
CVI_S32 CVI_IVE_DMA(IVE_HANDLE pIveHandle, IVE_DATA_S *pstSrc, IVE_DST_
↪DATA_S *pstDst, IVE_DMA_CTRL_S *pstCtrl, CVI_BOOL bInstant );
```

## 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 Copy 模式下不能为空。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。 True 为忙等待模式,False 为 中断模式。	输入

参数名称	支持类型	地址对齐	分辨率
pstSrc	IVE_DATA_S	1 byte	32x1~1920x1080
pstDst	IVE_DST_DATA_S	1 byte	直接拷贝时同 pst- Src; 间隔拷贝时比 pstSrc 大

## 【返回值】



返回值	描述
0	成功
非 0	失败

**【需求】**

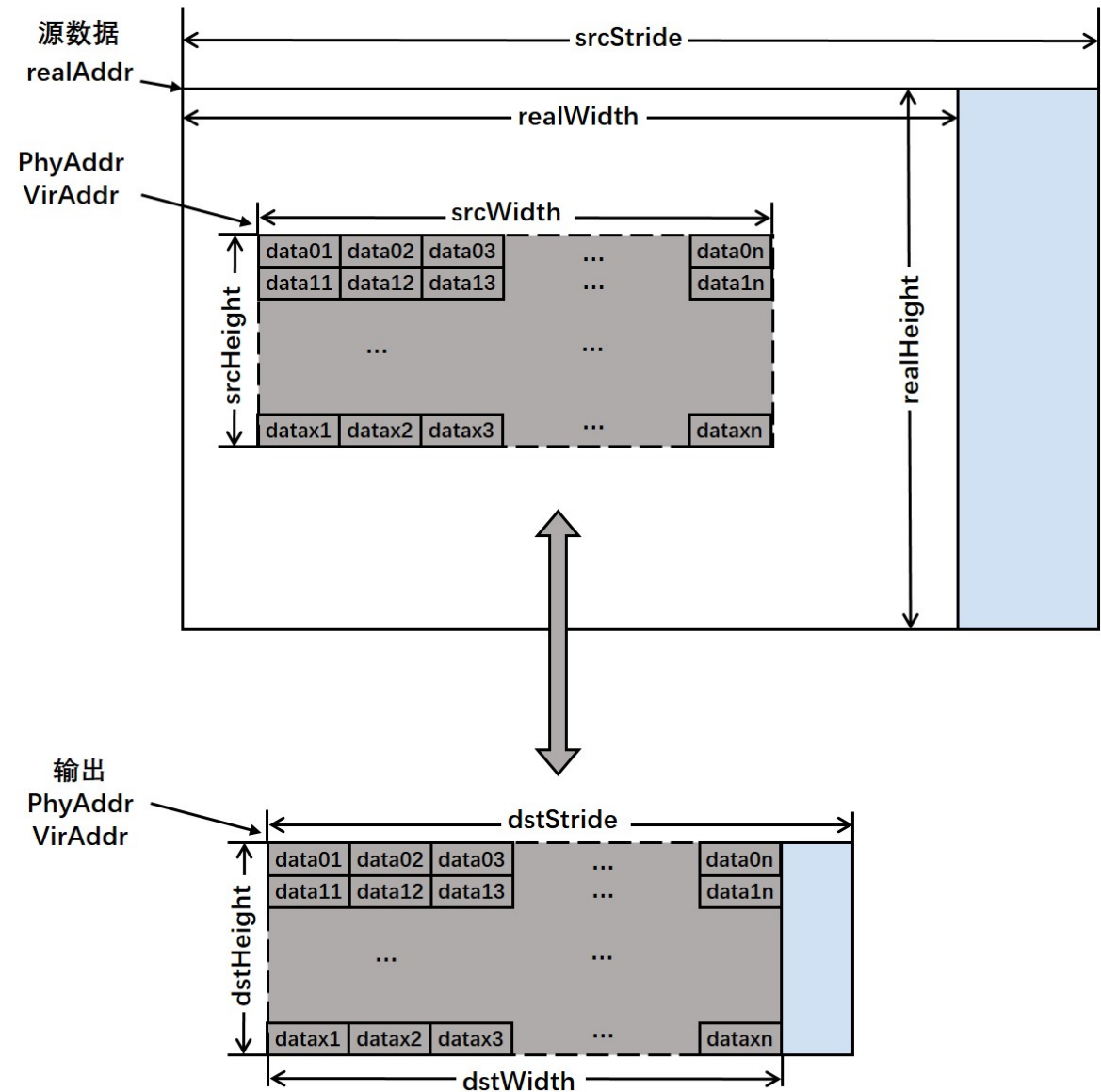
- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

**[注意]**

- `IVE_DMA_MODE_DIRECT_COPY`: 快速拷贝模式  
可实现从大块内存中扣取小块内存, 如图所示, 计算公式如下:

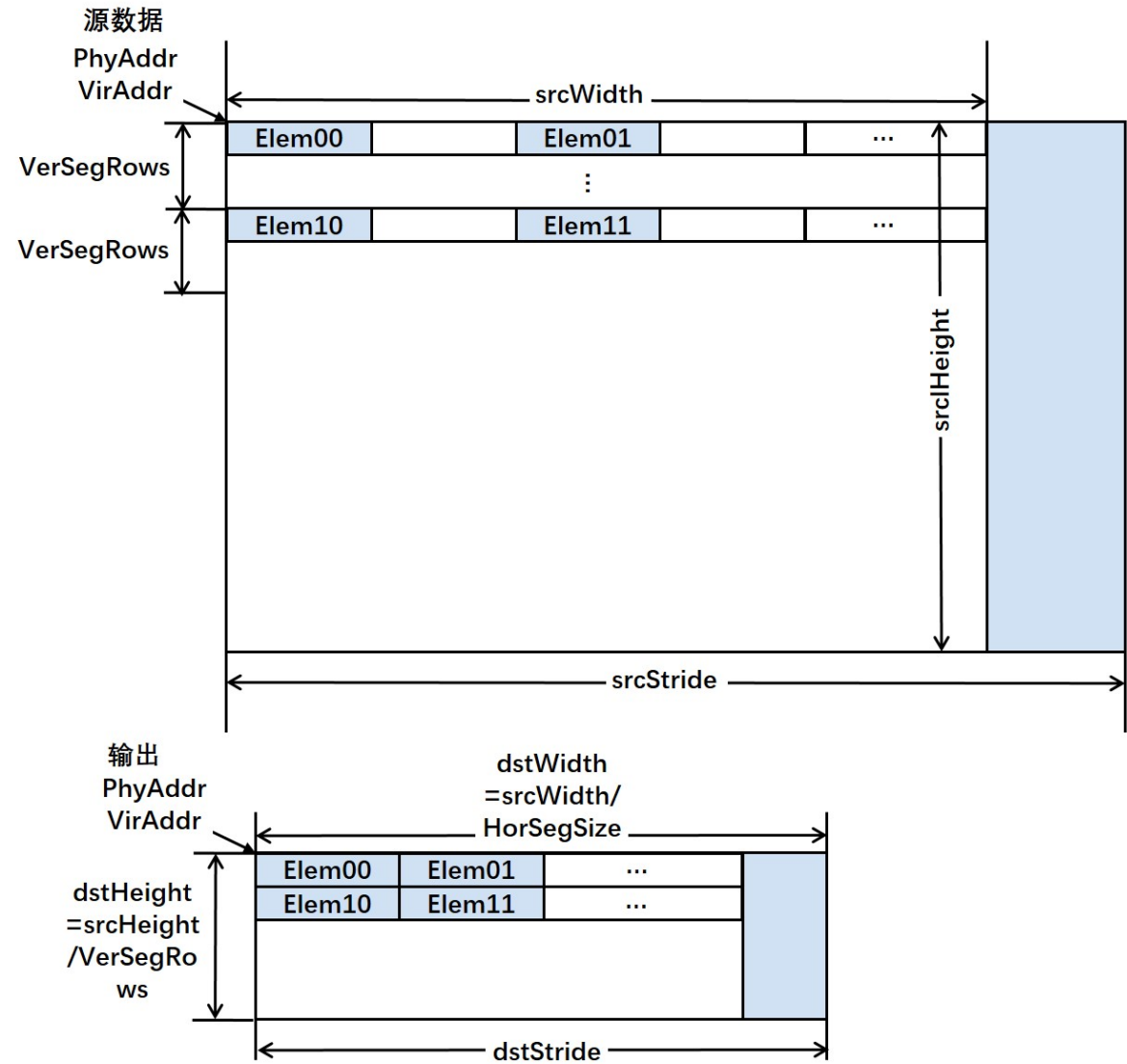
$$I_{\text{out}}(x, y) = I(x, y) \quad (0 \leq x \leq \text{width}, 0 \leq y \leq \text{height})$$

快速拷贝示意图



- IVE\_DMA\_MODE\_INTERVAL\_COPY: 间隔拷贝模式
  - 要求源数据宽度为 `u8HorSegSize` 的倍数;
  - 间隔拷贝的方式: 将每 `u8VerSegRows` 行中第一行数据分割为 `u8HorSegSize` 大小的段, 拷贝每段中的前 `u8ElemSize` 大小的字节。如图所示。
- IVE\_DMA\_MODE\_SET\_3BYTE: 3 字节填充模式
  - 仅使用 `pstSrc`, 用 `u64Val` 的低 3 字节对源数据进行填充操作; 当一行末尾不够 3 字节时, 用 `u64Val` 的低字节填充
- IVE\_DMA\_MODE\_SET\_8BYTE: 8 字节填充模式
  - 仅使用 `pstSrc`, 用 `u64Val` 对源数据进行填充操作; 当一行的末尾不足 8 字节时, 用 `u64Val` 的低字节填充。

间隔拷贝示意图



### 3.4 Filter

【描述】

创建 5x5 模板滤波任务，通过配置不同的模板系数，可以实现不同的滤波。

【语法】

```
CVI_S32 CVI_IVE_Filter(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_FILTER_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 宽、高同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1、YUV420SP、 YUV422SP	16 byte	64x64~1920x1080
pstDst	U8C1、YUV420SP、 YUV422SP	16 byte	同 pstSrc

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

[注意]

## 3.5 Filter And CSC

**【描述】**

创建 5x5 模板滤波和 CSC 图像格式转换任务，一次完成滤波和格式转换两种功能。

**【语法】**

```
CVI_S32 CVI_IVE_FilterAndCSC(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_IMAGE_S *pstDst, IVE_FILTER_AND_CSC_CTRL_S *pstCtrl, CVI_BOOL  
bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 宽、高同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	YUV420SP、 YUV422SP	16 byte	64x64~1920x1024
pstDst	U8C3_PLANAR、 U8C3_PACKAGE	16 byte	同 pstSrc

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

[注意]

## 3.6 CSC

**【描述】**

创建色彩空间转换任务, 可实现 YUV2RGB YUV2HSV RGB2YUV 的色彩空间转换。

**【语法】**

```
CVI_S32 CVI_IVE_CSC(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_CSC_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 宽、高同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	YUV420SP、 YUV422SP、 U8C3_PLANAR、 U8C3_PACKAGE	16 byte	64x64~1920x1080
pstDst	U8C3_PLANAR、 U8C3_PACKAGE、 YUV420SP、 YUV422SP	16 byte	同 pstSrc

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

·

[注意]

## 3.7 Sobel

**【描述】**

创建 5x5 模板 Sobel-like 梯度计算任务。

**【语法】**

```
CVI_S32 CVI_IVE_Sobel(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDstH, IVE_DST_IMAGE_S *pstDstV, IVE_SOBEL_CTRL_S
*pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDstH	由模板直接滤波得到的梯度分量图像 H 指针。根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。宽高同 pstSrc。	输出
pstDstV	由模板直接滤波得到的梯度分量图像 V 指针。根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。宽高同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	16x16~1920x1080
pstDstH	S16C1	16 byte	同 pstSrc
pstDstV	S16C1	16 byte	同 pstSrc

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

[注意]

## 3.8 NormGrad

### 【描述】

创建归一化梯度计算任务。所有梯度会归一化到 S8 格式。

### 【语法】

```
CVI_S32 CVI_IVE_NormGrad(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDstH, IVE_DST_IMAGE_S *pstDstV, IVE_DST_IMAGE_S
*pstDstHV, IVE_NORM_GRAD_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDstH	由模板直接滤波得到的梯度分量图像 H 指针。根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。 宽、高同 pstSrc。	输出
pstDstV	由模板直接滤波得到的梯度分量图像 V 指针。根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。 宽、高同 pstSrc。	输出
pstDstHV	由模板直接滤波得到的梯度分量图像 HV 指针, 采用 package 形式存储。依据 pstCtrl→enOutCtrl, 若需要输出则不能为空。 宽、高同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDstH	U8C1	16 byte	同 pstSrc
pstDstV	U8C1	16 byte	同 pstSrc
pstDstHV	S8C2_PACKAGE	16 byte	同 pstSrc

### 【返回值】



返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

.

[注意]

## 3.9 Canny Edge

**【描述】**

灰度图的 Canny 边缘提取的后半部: 连接边缘点, 形成 Canny 边缘图。

**【语法】**

```
CVI_S32 CVI_IVE_CannyEdge(IVE_IMAGE_S *pstEdge, IVE_MEM_INFO_S *pstStack);
```

**【参数】**

参数名称	描述	输入/输出
pstEdge	输入 Edge Flag 图像指针。 输出二值化边界图像指针。	输入/输出
pstStack	强边界的坐标	输入/输出

参数名称	支持图像类型	地址对齐	分辨率
pstEdge	U8C1	16 byte	64x64~1920x1080
pstStack		16 byte	

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

[注意]

## 3.10 Canny Hysteresis Edge

### 【描述】

创建 Canny Edge 任务，计算灰阶图像的梯度，梯度幅度，迟滞阈值和非极大值抑制。

### 【语法】

```
CVI_S32 CVI_IVE_CannyHysEdge(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstEdge, IVE_DST_MEM_INFO_S *pstStack,
IVE_CANNY_HYS_EDGE_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstEdge	强/弱 Edge Flag 图像。	输出
pstStack	强边界的坐标	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstEdge	U8C1	16 byte	同 pstSrc
pstStack		16 byte	

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h
- 

[注意]

## 3.11 MagAndAng

### 【描述】

创建 5x5 模板梯度幅值与幅角计算任务。

### 【语法】

```
CVI_S32 CVI_IVE_MagAndAng(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S
*pstSrc, IVE_DST_IMAGE_S *pstDstMag, IVE_DST_IMAGE_S *pstDstAng,
IVE_MAG_AND_ANG_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDstMag	输出幅值图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstDstAng	输出幅角图像指针。 根据 pstCtrl→enOutCtrl, 需要输出则不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDstMag	U16C1	16 byte	同 pstSrc
pstDstAng	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.12 Dilate

### 【描述】

创建二值图像 5x5 模板膨胀任务。

### 【语法】

```
CVI_S32 CVI_IVE_Dilate(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_IMAGE_S *pstDst, IVE_DILATE_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.13 Erode

### 【描述】

创建二值图像 5x5 模板腐蚀任务。

### 【语法】

```
CVI_S32 CVI_IVE_Erode(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_IMAGE_S *pstDst, IVE_ERODE_CTRL_S *pstErodeCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstErodeCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1024
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.14 Thresh

### 【描述】

创建灰度图像阈值化任务。

### 【语法】

```
CVI_S32 CVI_IVE_Thresh(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_IMAGE_S *pstDst, IVE_THRESH_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.15 And

### 【描述】

创建两二值图像相与任务。

### 【语法】

```
CVI_S32 CVI_IVE_And(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,  
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。	输出
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.16 Sub

### 【描述】

创建两灰度图像相减任务。

### 【语法】

```
CVI_S32 CVI_IVE_Sub(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,  
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst, IVE_SUB_CTRL_S *pstCtrl,  
CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。	输出
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc1。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc
pstDst	U8C1、S8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h



## 3.17 Or

### 【描述】

创建两二值图像相或任务。

### 【语法】

```
CVI_S32 CVI_IVE_Or(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。 高、宽同 pstSrc1。	输出
pstDst	输出幅值图像指针。 不能为空。 高、宽同 pstSrc1。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.18 Map

### 【描述】

将一个图像通过一个 256 维度的映射表格映射到另一个图像。

### 【语法】

```
CVI_S32 CVI_IVE_Map(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_SRC_MEM_INFO_S *pstMap, IVE_DST_IMAGE_S *pstDst, IVE_MAP_CTRL_S
*pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstMap	输入映射表格指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。高和宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstMap		16 byte	
pstDst	U8C1、U16C1、 S16C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.19 OrdStatFilter

### 【描述】

用 3x3 的核寻找图片中的极大、极小值。

### 【语法】

```
CVI_S32 CVI_IVE_OrdStatFilter(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_ORD_STAT_FILTER_CTRL_S *pstCtrl, CVI_BOOL
bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高和宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.20 Integral

### 【描述】

创建灰度图像的积分图计算任务。

### 【语法】

```
CVI_S32 CVI_IVE_Integ(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_MEM_INFO_S *pstDst, IVE_INTEG_CTRL_S *pstIntegCtrl, CVI_BOOL  
bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstIntegCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDst	U32C1, U64C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.21 Histogram

### 【描述】

创建灰度图像的直方图统计任务。

### 【语法】

```
CVI_S32 CVI_IVE_Hist(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_MEM_INFO_S *pstDst, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 不能为空。 高、宽同 pstSrc。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	64x64~1920x1080
pstDst		16 byte	

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.22 Add

### 【描述】

创建两灰度图像的加权加计算任务

### 【语法】

```
CVI_S32 CVI_IVE_Add(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst, IVE_ADD_CTRL_S
*pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。 高、宽同 pstSrc1	输入
pstDst	输出图像指针。 不能为空。 高、宽同 pstSrc1。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc1
pstDst	U8C1	16 byte	同 pstSrc1

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.23 Xor

### 【描述】

创建两二值图的异或计算任务

### 【语法】

```
CVI_S32 CVI_IVE_Xor(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,  
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。 高、宽同 pstSrc1	
pstDst	输出图像指针。 不能为空。 高、宽同 pstSrc1。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc1
pstDst	U8C1	16 byte	同 pstSrc1

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.24 Match BgModel

### 【描述】

基于 Codebook 演进的背景模型匹配。

### 【语法】

```
CVI_S32 CVI_IVE_MatchBgModel(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S
*pstCurImg, IVE_DATA_S *pstBgModel, IVE_IMAGE_S *pstFgFlag, IVE_DST_IMAGE_S
*pstDiffFg, IVE_DST_MEM_INFO_S *pstStatData, IVE_MATCH_BG_MODEL_CTRL_S
*pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstCurImg	当前图像指针。 不能为空。	输入
pstBgModel	模型指针。 不能为空。	输入/输出
pstFgFlag	前景状态图像指针。 不能为空。 高、宽同 pstCurImg。	输入/输出
pstDiffFg	差分前景图像指针。 不能为空。 高、宽同 pstCurImg。	输出
pstStatData	前景状态指针。 不能为空。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstCurImg	U8C1	16 byte	64x64~1920x1080
pstBgModel		16 byte	
pstFgFlag	U8C1	16 byte	同 pstCurImg
pstDiffFg	S8C1	16 byte	同 pstCurImg
pstStatData		16 byte	

### 【返回值】

返回值	描述
0	成功
非 0	失败



## 【需求】

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.25 Update BgModel

## 【描述】

基于 Codebook 演进的背景模型更新，对背景模型的内部状态进行更新。

## 【语法】

```
CVI_S32 CVI_IVE_UpdateBgModel(IVE_HANDLE pIveHandle,
IVE_SRC_IMAGE_S *pstCurImg, IVE_DATA_S *pstBgModel, IVE_IMAGE_S
*pstFgFlag, IVE_DST_IMAGE_S *pstBgImg, IVE_DST_IMAGE_S
*pstChaSta, IVE_DST_MEM_INFO_S *pstStatData, IVE_UPDATE_BG_MODEL_CTRL_S
*pstCtrl, CVI_BOOL bInstant);
```

## 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstCurImg	当前帧图像指针。 不能为空。	输入
pstBgModel	模型指针指针。 不能为空。	输入/输出
pstFgFlag	前景状态图像指针。 不能为空。	输入/输出
pstBgImg	背景图像指针。 不能为空。 高、宽同 pstFgFlag。	输出
pstChaSta	前景更新状态图像指针。	输出
pstStatData	背景状态指针。 不能为空。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstCurImg	U8C1	16 byte	64x64~1920x1080
pstBgModel		16 byte	
pstFgFlag	U8C1	16 byte	同 pstCurImg
pstBgImg	U8C1	16 byte	
pstChaSta	S8C1	16 byte	同 pstCurImg
pstStatData		16 byte	

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.26 Gradient of Foreground

**【描述】**

根据背景梯度图像和当前图像，计算前景图梯度图像。

**【语法】**

```
CVI_S32 CVI_IVE_GradFg(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstBgDiffFg,  
IVE_SRC_IMAGE_S *pstCurGrad, IVE_SRC_IMAGE_S *pstBgGrad, IVE_DST_IMAGE_S  
*pstGradFg, IVE_GRAD_FG_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstBgDiffFg	前景图像指针。 不能为空。	输入
pstCurGrad	当前梯度图像指针。 不能为空。	输入
pstBgGrad	背景梯度图像指针。 不能为空。 高、宽同 pstCurGrad。	输入
pstGradFg	前景梯度图像。 不能为空。 高、宽同 pstCurGrad。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstBgDiffFg	S8C1	16byte	64x64~1920x1080
pstCurGrad	S8C2_PACKAGE	16byte	同 pstBgDiffFg
pstBgGrad	S8C2_PACKAGE	16byte	同 pstBgDiffFg
pstGradFg	U8C1	16byte	同 pstBgDiffFg

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

· 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.27 GMM

**【描述】**

建立 GMM 背景模型任务。

**【语法】**

```
CVI_S32 CVI_IVE_GMM(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstFg, IVE_DST_IMAGE_S *pstBg, IVE_MEM_INFO_S
*pstModel, IVE_GMM_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstFg	前景图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstBg	前景图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstModel	模型指针指针。 不能为空。	输入/输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstModel		16byte	
pstSrc	U8C1	16byte	64x64~1920x1080
pstFg	U8C1	16byte	同 pstSrc
pstBg	U8C1	16byte	同 pstBgDiffFg

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.28 GMM2

**【描述】**

建立 GMM2 背景模型任务。

**【语法】**

```
CVI_S32 CVI_IVE_GMM2(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_SRC_IMAGE_S *pstFactor, IVE_DST_IMAGE_S *pstFg, IVE_DST_IMAGE_S
```

(下页继续)

(续上页)

```
*pstBg, IVE_DST_IMAGE_S *pstMatchModelInfo, IVE_MEM_INFO_S *pstModel,
IVE_GMM2_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

## 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstFactor	模型更新系数指针。 当且仅当 pstCtrl→enSnsFactorMode、 pstCtrl→enLifeUpdateFactorMode 均使用全局模式时可以为空。	输入
pstFg	前景图像指针。 不能为空。高、宽同 pstSrc。	输出
pstBg	前景图像指针。 不能为空。高、宽同 pstSrc。	输出
pstMatchModelInfo	模型匹配系数指针。 不能为空。	输出
pstModel	模型指针指针。 不能为空。	输入/输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstModel	.	16byte	
pstFactor	U16C1	16byte	同 pstSrc
pstSrc	U8C1、 U8C3_PACKAGE	16byte	64x64~1920x1080
pstFg	U8C1 二值化图像	16byte	同 pstSrc。
pstBg	U8C1、 U8C3_PACKAGE	16byte	同 pstSrc。
pstMatchModelInfo	U8C1	16byte	同 pstSrc。

## 【返回值】

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.29 Bernsen

**【描述】**

建立 Bernsen 二值化算法任务。

**【语法】**

```
CVI_S32 CVI_IVE_Bernsen(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,  
IVE_DST_IMAGE_S *pstDst, IVE_BERNSEN_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	16x16~1920x1080
pstDst	U8C1	16 byte	同 pstSrc

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.30 NCC

### 【描述】

创建两相同分辨率灰度图像的归一化互相关系数计算任务

### 【语法】

```
CVI_S32 CVI_IVE_NCC(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_MEM_INFO_S *pstDst, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。 高、宽同 pstSrc1。	输入
pstDst	输出相关系数指针。 不能为空。 内存至少需配置:sizeof(IVE_NCC_DST_MEM_S)。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc1
pstDst	.	16byte	.

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.31 CCL

### 【描述】

创建二值图像的连通区域标记任务。

### 【语法】

```
CVI_S32 CVI_IVE_CCL(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrcDst,  
IVE_DST_MEM_INFO_S *pstBlob, IVE_CCL_CTRL_S *pstCclCtrl, CVI_BOOL  
bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrcDst	输入图像指针。 不能为空。	输入
pstBlob	连通区域信息指针。 不能为空。 内存至少需配置为 sizeof (IVE_CCBLOB_S) 大小, 最多输出 254 个有效的连通 区域。	输入
pstCclCtrl	控制结构体指针。 不能为空。	输出
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1	16byte	64x64~1920x1080
pstBlob		16byte	

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h



## 3.32 LBP

### 【描述】

创建 LBP 计算任务

### 【语法】

```
CVI_S32 CVI_IVE_LBP(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_LBP_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输出图像指针。 不能为空。高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16byte	64x64~1920x1024
pstDst	U8C1	16 byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

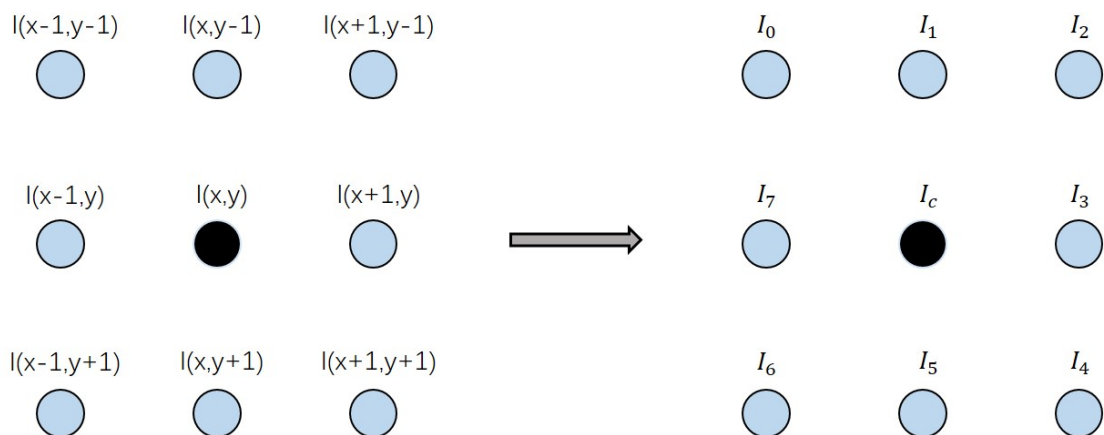
### 【需求】

- 头文件：cvi\_comm\_ive.h cvi\_ive.h

[注意]

LBP 计算公式如图所示：

LBP 计算公式示意图



· IVE\_LBP\_CMP\_MODE\_NORMAL

$$lbp(x, y) = \sum_{i=0}^7 ((I_i - I_c) \geq thr) \ll (7 - i), \text{ 其中 } thr \in [-128, 127];$$

· IVE\_LBP\_CMP\_MODE\_ABS

$$lbp(x, y) = \sum_{i=0}^7 (abs(I_i - I_c) \geq thr) \ll (7 - i), \text{ 其中 } thr \in [0, 255];$$

其中,  $I(x, y)$  对应 input,  $lbp(x, y)$  对应 output,  $thr$  对应 `lbp_attr->un8BitThr`。

### 3.33 SAD

#### 【描述】

计算两幅图像 SAD。

#### 【语法】

```
CVI_S32 CVI_IVE_SAD(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc1,
IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstSad, IVE_DST_IMAGE_S
*pstThr, IVE_SAD_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

#### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。 高、宽同 pstSrc1	输入
pstSad	输出 SAD 图像。 根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。 根据 pstCtrl→enMode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pstSrc1 的 1/4、1/8、1/16。	输出
pstThr	输出 SAD 阈值化图像指针。 根据 pstCtrl→enOutCtrl, 若需要输出则不能为空。 根据 pstCtrl→enMode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pstSrc1 的 1/4、1/8、1/16。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc1
pstSad	U8C1、U16C1	16 byte	根据 pstCtrl→enMode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pstSrc1 的 1/4、1/8、1/16。
pstThr	U8C1	16 byte	根据 pstCtrl→enMode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pstSrc1 的 1/4、1/8、1/16。

## 【返回值】

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.34 BufFlush

**【描述】**

对于使用 `CVI_IVE_CreateImage_Cached` 建立的图像, 在 IVE 硬件存取图像的内容前, 须使用此函数将 cache data 放到 RAM。

**【语法】**

```
CVI_S32 CVI_IVE_BufFlush(IVE_HANDLE pIveHandle, IVE_IMAGES_S *pstImg);
```

**【参数】**

参数名称	描述	输入/输出
<code>pIveHandle</code>	handle 指针。 不能为空。	输入
<code>pstImg</code>	操作图像指针。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.35 BufRequest

**【描述】**

对于使用 `CVI_IVE_CreateImage_Cached` 建立的图像, 在处理器存取 `u64VirAddr` 所指向的内容前, 须使用此函数将 Ram 内容更新到 cache。

**【语法】**

```
CVI_S32 CVI_IVE_BufRequest(IVE_HANDLE pIveHandle, IVE_IMAGES_S *pstImg);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImg	操作的图像。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.36 CreateMemInfo

**【描述】**

创造一块内存供 IVE\_MEM\_S 结构使用。

**【语法】**

```
CVI_S32 CVI_IVE_CreateMemInfo(IVE_HANDLE pIveHandle, IVE_MEM_INFO_S  
*pstMemInfo, CVI_U32 u32ByteSize);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstMemInfo	内存结构体指针。 不能为空。	输入
u32ByteSize	内存结构体的 byte 容量。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.37 CreateDataInfo

**【描述】**

创造一块内存供 IVE\_DATA\_S 结构使用。

**【语法】**

```
CVI_S32 CVI_IVE_CreateDataInfo(IVE_HANDLE pIveHandle, IVE_DATA_S *pstDataInfo, CVI_U16 u16Width, CVI_U16 u16Height);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstDataInfo	数据信息指针。 不能为空。	输入
u16Width	数据的宽。	输入
u16Height	数据的高。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.38 CreateImage

**【描述】**

创造一块图像内存供使用。用此函数建立的图像会自动映射 u64PhyAddr 和 u64VirAddr 的内容。无需对 cache 进行 Flush 或 Invalidate。

**【语法】**

```
CVI_S32 CVI_IVE_CreateImage(IVE_HANDLE pIveHandle, IVE_IMAGE_S *pstImg,  
IVE_IMAGE_TYPE_E enType, uint16_t u16Width, uint16_t u16Height);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImg	输出图像指针。	输出
enType	图像格式。	输入
u16Width	图像的宽。	输入
u16Height	图像的高。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.39 CreateImage with Cache

**【描述】**

创造一块图像内存供使用。用此函数建立的图像需要使用 CVI\_IVE\_BufFlush 和 CVI\_IVE\_BufRequest 更新 u64PhyAddr 和 u64VirAddr 的内容。

**【语法】**

```
CVI_S32 CVI_IVE_CreateImage_Cached(IVE_HANDLE pIveHandle, IVE_IMAGE_S  
*pstImg, IVE_IMAGE_TYPE_E enType, uint16_t u32Width, uint16_t u32Height);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImg	输出图像指针。	输出
enType	图像格式。	输入
u32Width	图像的宽	输入
u32Height	图像的高	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.40 ResetImage

**【描述】**

将 Image 内容填入特定值。

**【语法】**

```
CVI_S32 CVI_IVE_ResetImage(IVE_IMAGE_S *pstImg, CVI_U8 val);
```

**【参数】**

参数名称	描述	输入/输出
<code>pstImg</code>	输出图像指针。	输出
<code>val</code>	预填入图像的值。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.41 ReadImageArray

**【描述】**

从 buffer 读入图像。

**【语法】**



```
CVI_S32 CVI_IVE_ReadImageArray(IVE_HANDLE pIveHandle, IVE_IMAGE_S
*pstImage, char *pBuffer, IVE_IMAGE_TYPE_E enType, CVI_U16 u16Width,
CVI_U16 u16Height);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImage	输出图像指针。	输出
pBuffer	Buffer 指针。	输入
enType	图像格式。	输入
u16Width	图像的宽。	输入
u16Height	图像的高。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.42 ReadMem

**【描述】**

从文件读取到 IVE\_DATA\_S 结构。

**【语法】**

```
CVI_S32 CVI_IVE_ReadMem(IVE_HANDLE pIveHandle, IVE_MEM_INFO_S *pstMem,
const char *filename, CVI_U32 uSize);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstMem	内存结构体指针。	输出
filename	文件路径。	输入
u32Size	Mem 大小。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.43 ReadMemArray

**【描述】**

从 buffer 读入数据到 IVE\_MEM\_INFO\_S 结构。

**【语法】**

```
CVI_S32 CVI_IVE_ReadMemArray (IVE_HANDLE pIveHandle, IVE_MEM_INFO_S *pstMem,
char *pBuffer, CVI_U32 uSize);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstMem	内存结构体指针。	输出
pBuffer	Buffer 指针	输入
u32Size	Buffer 大小。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.44 ReadData

### 【描述】

从文件读入到 IVE\_DATA\_S 结构。

### 【语法】

```
CVI_S32 CVI_IVE_ReadData(IVE_HANDLE pIveHandle, IVE_DATA_S *pstData,  
const char *filename, CVI_U16 u16Width, CVI_U16 u16Height);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstData	数据结构体指针。	输出
filename	文件路径。	输入
u16Width	数据的宽。	输入
u16Height	数据的高。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.45 ReadDataArray

### 【描述】

从 buffer 读入数据到 IVE\_DATA\_S 结构。

### 【语法】

```
CVI_S32 CVI_IVE_ReadDataArray(IVE_HANDLE pIveHandle, IVE_DATA_S *pstData,  
char *pBuffer, CVI_U16 u16Width, CVI_U16 u16Height);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstData	数据结构体指针。	输出
pBuffer	Buffer。	输入
u16Width	数据的宽。	输入
u16Height	数据的高。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.46 ReadImage

**【描述】**

从档案位置读取一张图像。

**【语法】**

```
IVE_IMAGE_S CVI_IVE_ReadImage(IVE_HANDLE pIveHandle, const char *filename,  
IVE_IMAGE_TYPE_E enType);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
filename	文件路径。	输入
enType	图像格式	

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

· 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.47 ReadRawImage

### 【描述】

从文件读取一张 Raw Image。

### 【语法】

```
IVE_IMAGE_S CVI_IVE_ReadRawImage(IVE_HANDLE pIveHandle, IVE_IMAGE_S *pstImg,  
const char *filename, IVE_IMAGE_TYPE_E enType, CVI_U16 u16Width, CVI_U16  
u16Height);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImg	输出图像指针。	输出
filename	文件路径。	输入
enType	图像格式。	输入
u16Width	图像的宽。	输入
u16Height	图像的高。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

· 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.48 WriteData

### 【描述】

写入 IVE\_DATA\_S 内容到文件。

### 【语法】

```
CVI_S32 CVI_IVE_WriteData(IVE_HANDLE pIveHandle, const char *filename,  
IVE_DATA_S *pstData);
```

## 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 的指针。 不能为空。	输入
filename	文件路径。	输入
pstData	要储存的内容。	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败

## 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.49 WriteMem

## 【描述】

写入 IVE\_MEM\_INFO\_S 内容到文件位置。

## 【语法】

```
CVI_S32 CVI_IVE_WriteMem(IVE_HANDLE pIveHandle, const char *filename,  
IVE_MEM_INFO_S *pstMem);
```

## 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 的指针。 不能为空。	输入
filename	文件路径。	输入
pstMem	要储存的内容。	输出

## 【返回值】

返回值	描述
0	成功
非 0	失败

## 【需求】

· 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.50 WriteImage

### 【描述】

写入一张 PNG 图像到文件。

### 【语法】

```
CVI_S32 CVI_IVE_WriteImage(IVE_HANDLE pIveHandle, const char *filename,
IVE_IMAGE_S *pstImg);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
filename	文件路径。	输入
pstImg	输出图像指针。	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

· 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.51 WriteRawImage

### 【描述】

写入一张图像到文件。

### 【语法】

```
CVI_S32 CVI_IVE_WriteRawImage(IVE_HANDLE pIveHandle, const char *filename,
IVE_IMAGE_S *pstImg);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
filename	文件路径。	输入
pstImg	输出图像指针。	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.52 Reset Register

**【描述】**

重置 IVE 的缓存器为默认值。

**【语法】**

```
CVI_S32 CVI_IVE_RESET(IVE_HANDLE pIveHandle, int select);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
select	要重置的 IVE Module。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h



## 3.53 Dump Register

### 【描述】

输出 IVE 的缓存器值到 Log。

### 【语法】

```
CVI_S32 CVI_IVE_DUMP(IVE_HANDLE pIveHandle);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.54 Split DiffFg of BgModel

### 【描述】

从 BgModel 的结果取出 DiffFg，并储存成 YUV 图像。

### 【语法】

```
CVI_S32 CVI_IVE_DiffFg_Split(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstDiffFg, IVE_DST_IMAGE_S *pstBGDiffFg, IVE_DST_IMAGE_S *pstFrmDiffFg);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstDiffFg		输入
pstBGDiffFg		输出
pstFrmDiffFg		输出

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.55 Split ChgSta of BgModel

**【描述】**

从 BgModel 的结果取出 ChgSta, 并储存成 YUV 图像。

**【语法】**

```
CVI_S32 CVI_IVE_ChgSta_Split(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S
*pstChgSta, IVE_DST_IMAGE_S *pstChgStaImg, IVE_DST_IMAGE_S *pstChgStaFg,
IVE_DST_IMAGE_S *pstChStaLift);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstChgSta		输入
pstChgStaImg		输出
pstChgStaFg		输出
pstChStaLift		输出

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.56 Query Tasks

### 【描述】

查询现有 Task 状态。

### 【语法】

```
CVI_S32 CVI_IVE_QUERY(IVE_HANDLE pIveHandle, CVI_BOOL *pbFinish,  
CVI_BOOL bBlock);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pbFinish	任务结束标志。	输出
bBlock	任务阻塞标志。	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

头文件：cvi\_comm\_ive.h cvi\_ive.h

## 3.57 Image2VideoFrameInfo

### 【描述】

将 IVE 图像格式转换成 Video Frame Info 格式。

### 【语法】

```
CVI_S32 CVI_IVE_Image2VideoFrameInfo(IVE_IMAGE_S *pstIISrc,  
VIDEO_FRAME_INFO_S *pstVFIDst);
```

### 【参数】

参数名称	描述	输入/输出
pstIISrc	输入图像指针。	输入
pstVFIDst	输出图像指针。	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.58 VideoFrameInfo2Image

**【描述】**

将 Video Frame Info 图像格式转换成 IVE 格式。

**【语法】**

```
CVI_S32 CVI_IVE_VideoFrameInfo2Image(VIDEO_FRAME_INFO_S *pstVFISrc,  
IVE_IMAGE_S *pstIIDst);
```

**【参数】**

参数名称	描述	输入/输出
<code>pstIISrc</code>	输入图像指针。	输入
<code>pstVFIDst</code>	输出图像指针。	输出

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: `cvi_comm_ive.h` `cvi_ive.h`

## 3.59 FreeM

### 【描述】

释放一块 IVE\_MEM\_INFO\_S 结构。

### 【语法】

```
CVI_S32 CVI_SYS_FreeM(IVE_HANDLE pIveHandle, IVE_MEM_INFO_S *pstMem);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstMem	要释放的内存结构体指针。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.60 FreeI

### 【描述】

释放一张 IVE\_IMAGE\_S 结构。

### 【语法】

```
CVI_S32 CVI_SYS_FreeI(IVE_HANDLE pIveHandle, IVE_IMAGE_S *pstImg);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstImg	输入图像指针。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.61 FreeD

**【描述】**

释放 IVE\_DATA\_S 结构。

**【语法】**

```
CVI_S32 CVI_SYS_FreeD(IVE_HANDLE pIveHandle, IVE_DATA_S *pstData);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstData	输入数据指针。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

·

## 3.62 Thresh\_S16

### 【描述】

创建 S16 数据到 8bit 数据的阈值化任务。

### 【语法】

```
CVI_S32 CVI_IVE_Thresh_S16(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_THRESH_S16_CTRL_S *pstThrS16Ctrl,
CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输出

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	S16C1	2byte	64x64~1920x1080
pstDst	U8C1、S8C1	1byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.63 Thresh\_U16

### 【描述】

创建 U16 数据到 8bit 数据的阈值化任务。

### 【语法】

```
CVI_S32 CVI_IVE_Thresh_U16(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc, IVE_DST_IMAGE_S *pstDst, IVE_THRESH_U16_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输出

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U16C1	2byte	64x64~1920x1080
pstDst	U8C1	1byte	同 pstSrc

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

· 头文件: cvl\_comm\_ive.h cvl\_ive.h

·



## 3.64 Resize

### 【描述】

创建图像 Resize 任务，支持 Bilinear Interpolation 及 Area Interpolation 方法。

### 【语法】

```
CVI_S32 CVI_IVE_Resize(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S astSrc[],  
IVE_DST_IMAGE_S astDst[], IVE_RESIZE_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
astSrc	输入图像数组。 不能为空。	输入
astDst	输出的图像数组。 不能为空。 图像类型必须和 astSrc 相同	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输出

参数名称	支持图像类型	地址对齐	分辨率
astSrc	U8C1、 U8C3_PLANAR	16byte	4x4~1920x1080
astDst	U8C1、 U8C3_PLANAR	16byte	4x4~1920x1080

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件：cvi\_comm\_ive.h cvi\_ive.h

## 3.65 16BitTo8Bit

### 【描述】

创建 16bit 图像数据到 8bit 图像数据的线性化任务。

### 【语法】

```
CVI_S32 CVI_IVE_16BitTo8Bit (IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc,
IVE_DST_IMAGE_S *pstDst, IVE_16BIT_TO_8BIT_CTRL_S *pstCtrl, CVI_BOOL
bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针 不能为空。	输入
bInstant	及时返回结果标志。	输出

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U16C1、S16C1	2byte	16x16~1920x1080
pstDst	U8C1、S8C1	1byte	16x16~1920x1080

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.66 RGB YUV Erode to Dilate

### 【描述】

创建 RGB 图像转化为 YUV 图像的腐蚀膨胀任务。

### 【语法】

```
CVI_S32 CVI_IVE_rgbPToYuvToErodeToDilate(IVE_HANDLE pIveHandle,  
IVE_SRC_IMAGE_S *pstSrc, IVE_DST_IMAGE_S *pstDst1, IVE_DST_IMAGE_S *pstDst2,  
IVE_FILTER_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst1	输出 1 图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstDst2	输出 2 图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.67 STCandiCorner

### 【描述】

计算候选角点。

### 【语法】

```
CVI_S32 CVI_IVE_STCandiCorner(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S *pstSrc, IVE_DST_IMAGE_S *pstDst, IVE_ST_CANDI_CORNER_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

### 【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	参考值	输出

### 【返回值】

返回值	描述
0	成功
非 0	失败

### 【需求】

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.68 Background Subtraction

### 【描述】

创建背景相减法任务。

### 【语法】

```
CVI_S32 CVI_IVE_FrameDiffMotion(IVE_HANDLE pIveHandle, IVE_SRC_IMAGE_S
*pstSrc1, IVE_SRC_IMAGE_S *pstSrc2, IVE_DST_IMAGE_S *pstDst,
IVE_FRAME_DIFF_MOTION_CTRL_S *pstCtrl, CVI_BOOL bInstant);
```

**【参数】**

参数名称	描述	输入/输出
pIveHandle	handle 指针。 不能为空。	输入
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。	输入
pstDst	输入图像指针。 不能为空。 高、宽同 pstSrc。	输出
pstCtrl	控制结构体指针。 不能为空。	输入
bInstant	及时返回结果标志。	输出

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	16 byte	64x64~1920x1080
pstSrc2	U8C1	16 byte	同 pstSrc1
pstDst	U8C1	16 byte	同 pstSrc1

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.69 Compare Ive Image

**【描述】**

创建 IVE 图像对比任务。

**【语法】**

```
CVI_S32 CVI_IVE_CompareIveImage(IVE_IMAGE_S *pstImage1,  
    IVE_IMAGE_S *pstImage2);
```

**【参数】**

参数名称	描述	输入/输出
pstSrc1	输入图像指针。 不能为空。	输入
pstSrc2	输入图像指针。 不能为空。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.70 Compare Ive Mem

**【描述】**

创建 IVE 内存对比任务。

**【语法】**

```
CVI_S32 CVI_IVE_CompareIveMem(IVE_MEM_INFO_S *pstMem1,  
    IVE_MEM_INFO_S *pstMem2);
```

**【参数】**

参数名称	描述	输入/输出
pstMem1	输入内存指针。 不能为空。	输入
pstMem2	输入内存指针。 不能为空。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.71 Compare Ive Data

**【描述】**

创建 IVE 数据对比任务。

**【语法】**

```
CVI_S32 CVI_IVE_CompareIveData(IVE_DATA_S *pstData1,  
    IVE_DATA_S *pstData2);
```

**【参数】**

参数名称	描述	输入/输出
pstData1	输入数据指针。 不能为空。	输入
pstData2	输入数据指针。 不能为空。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h

## 3.72 Compare Ive SAD Image

**【描述】**

创建 IVE SAD 图像对比任务。

**【语法】**

```
CVI_S32 CVI_IVE_CompareSADImage(IVE_IMAGE_S *pstImage1,  
    IVE_IMAGE_S *pstImage2, IVE_SAD_MODE_E mode, CVI_BOOL isDMAhalf);
```

**【参数】**

参数名称	描述	输入/输出
pstImage1	输入数据指针。 不能为空。	输入
pstImage2	输入数据指针。 不能为空。	输入
mode	SAD 模式。 不能为空。	输入
isDMAhalf	是否输出一半结果。 不能为空。	输入

**【返回值】**

返回值	描述
0	成功
非 0	失败

**【需求】**

- 头文件: cvi\_comm\_ive.h cvi\_ive.h



# 4 数据类型和数据结构

---

IVE 相关数据类型及数据结构定义描述如下：

- IVE\_IMAGE\_TYPE\_E: 定义二维广义图像支持的图像类型。
- IVE\_IMAGE\_S: 定义二维广义图像信息。
- IVE\_SRC\_IMAGE\_S: 定义源图像。
- IVE\_DST\_IMAGE\_S: 定义输出图像。
- IVE\_DATA\_S: 定义以 byte 为单位的二维图像信息。
- IVE\_SRC\_DATA\_S: 定义以 byte 为单位的二维源数据信息。
- IVE\_DST\_DATA\_S: 定义 byte 为单位的二维输出数据信息。
- IVE\_MEM\_INFO\_S: 定义一维数据内存信息。
- IVE\_SRC\_MEM\_INFO\_S: 定义一维源数据。
- IVE\_DST\_MEM\_INFO\_S: 定义一维输出数据。
- IVE\_8BIT\_U: 定义 8bit 数据共享体。
- IVE\_POINT\_U16\_S: 定义 unsigned 16bit 坐标数据结构体。
- IVE\_POINT\_S16\_S: 定义 signed 16bit 坐标数据结构体。
- IVE\_RECT\_U16\_S: 定义 unsigned 16bit 矩形数据结构体。
- IVE\_LOOK\_UP\_TABLE\_S: 定义 LUT 表格结构体。
- IVE\_DMA\_MODE\_E: 定义 DMA 运算模式。
- IVE\_DMA\_CTRL\_S: 定义 DMA 控制信息。
- IVE\_FILTER\_CTRL\_S: 定义模板滤波控制信息。
- IVE\_CSC\_MODE\_E: 定义色彩空间转换模式。
- IVE\_CSC\_CTRL\_S: 定义色彩空间转换控制信息。
- IVE\_SOBEL\_OUT\_CTRL\_E: 定义 sobel 输出控制信息。
- IVE\_SOBEL\_CTRL\_S: 定义 sobel 边缘提取控制信息。

- IVE\_MAG\_AND\_ANG\_OUT\_CTRL\_E: 定义幅值与角度计算的输出格式。
- IVE\_MAG\_AND\_ANG\_CTRL\_S: 定义幅值和幅角计算的控制信息。
- IVE\_DILATE\_CTRL\_S: 定义膨胀控制信息。
- IVE\_ERODE\_CTRL\_S: 定义腐蚀控制信息。
- IVE\_BLOCK\_CTRL\_S: 定义 IVE\_BLOCK 控制信息。
- IVE\_SUB\_MODE\_E: 定义两图像相减输出格式。
- IVE\_SUB\_CTRL\_S: 定义两图像相减控制参数。
- IVE\_INTEG\_OUT\_CTRL\_E: 定义积分图输出控制参数。
- IVE\_INTEG\_CTRL\_S: 定义积分图计算控制参数。
- IVE\_THRESH\_MODE\_E: 定义图像二值化输出格式。
- IVE\_THRESH\_CTRL\_S: 定义图像二值化控制信息。
- IVE\_THRESH\_S16\_MODE\_E: 定义 16bit 有符号图像的阈值化模式。
- IVE\_THRESH\_S16\_CTRL\_S: 定义 16bit 有符号图像的阈值化控制参数。
- IVE\_THRESH\_U16\_MODE\_E: 定义 16bit 无符号图像的阈值化模式。
- IVE\_THRESH\_U16\_CTRL\_S: 定义 16bit 无符号图像的阈值化控制参数。
- IVE\_16BIT\_TO\_8BIT\_MODE\_E: 定义 16bit 图像到 8bit 图像的转化模式。
- IVE\_16BIT\_TO\_8BIT\_CTRL\_S: 定义 16bit 图像到 8bit 图像的转化控制参数。
- IVE\_ORD\_STAT\_FILTER\_MODE\_E: 定义顺序统计量滤波模式。
- IVE\_ORD\_STAT\_FILTER\_CTRL\_S: 定义顺序统计量滤波控制参数。
- IVE\_MAP\_MODE\_E: 定义 MAP 的模式。
- IVE\_MAP\_CTRL\_S: 定义 MAP 的控制参数。
- IVE\_MAP\_U8BIT\_LUT\_MEM\_S: 定义 MAP 8bit 无符号查找表的内存。
- IVE\_MAP\_U16BIT\_LUT\_MEM\_S: 定义 MAP 16bit 无符号查找表的内存。
- IVE\_MAP\_S16BIT\_LUT\_MEM\_S: 定义 MAP 8bit 有符号查找表的内存。
- IVE\_EQUALIZE\_HIST\_CTRL\_MEM\_S: 定义直方图均衡化控制参数内存。
- IVE\_EQUALIZE\_HIST\_CTRL\_S: 定义直方图均衡化控制参数。
- IVE\_ADD\_CTRL\_S: 定义两图像的加权加控制参数。
- IVE\_NCC\_DST\_MEM\_S: 定义 NCC 的输出内存信息。
- IVE\_CANNY\_HYS\_EDGE\_CTRL\_S: 定义 CANNY 边缘算子控制参数。
- IVE\_CANNY\_STACK\_SIZE\_S: 定义 CANNY 边缘算子栈空间大小。
- IVE\_LBP\_CMP\_MODE\_E: 定义 LBP 纹理计算控制参数。

- IVE\_LBP\_CTRL\_S: 定义 LBP 纹理计算控制参数。
- IVE\_NORM\_GRAD\_OUT\_CTRL\_E: 定义归一化梯度信息计算任务输出控制枚举类型。
- IVE\_NORM\_GRAD\_CTRL\_S: 定义归一化梯度信息计算控制参数。
- IVE\_ST\_CANDI\_CORNER\_CTRL\_S: 定义角点检测控制参数。
- IVE\_ST\_CORNER\_INFO\_S: 定义角点信息。
- IVE\_ST\_CORNER\_CTRL\_S: 定义角点控制参数。
- IVE\_GRAD\_FG\_MODE\_E: 定义 GRAD FG 模式。
- IVE\_FG\_STAT\_DATA\_S: 定义前景数据。
- IVE\_BG\_STAT\_DATA\_S: 定义背景数据。
- IVE\_SAD\_MODE\_E: 定义 SAD 计算模式。
- IVE\_SAD\_OUT\_CTRL\_E: 定义 SAD 输出控制模式。
- IVE\_SAD\_CTRL\_S: 定义 SAD 控制参数。
- IVE\_RESIZE\_MODE\_E: 定义 Resize 的模式。
- IVE\_RESIZE\_CTRL\_S: 定义 Resize 控制参数。
- IVE\_REGION\_S: 定义连通区域信息。
- IVE\_CCBLOB\_S: 定义连通区域标记的输出信息。
- IVE\_CCL\_CTRL\_S: 定义连通区域标记控制参数。
- IVE\_CCL\_MODE\_E: 定义连通区域模式。

## 4.1 定义数据类型

### 【说明】

定义定点化的数据类型。

### 【定义】

与 middleware 共享，详见 `cvi_type.h`。

## 4.2 定义结构类型

### 4.2.1 IVE\_IMAGE\_TYPE\_E

**【说明】**

定义二维广义图像支持的图像类型。

**【定义】**

```
typedef enum _IVE_IMAGE_TYPE_E {  
    IVE_IMAGE_TYPE_U8C1 = 0x0,  
    IVE_IMAGE_TYPE_S8C1 = 0x1,  
    IVE_IMAGE_TYPE_YUV420SP = 0x2,  
    IVE_IMAGE_TYPE_YUV422SP = 0x3,  
    IVE_IMAGE_TYPE_YUV420P = 0x4,  
    IVE_IMAGE_TYPE_YUV422P = 0x5,  
    IVE_IMAGE_TYPE_S8C2_PACKAGE = 0x6,  
    IVE_IMAGE_TYPE_S8C2_PLANAR = 0x7,  
    IVE_IMAGE_TYPE_S16C1 = 0x8,  
    IVE_IMAGE_TYPE_U16C1 = 0x9,  
    IVE_IMAGE_TYPE_U8C3_PACKAGE = 0xa,  
    IVE_IMAGE_TYPE_U8C3_PLANAR = 0xb,  
    IVE_IMAGE_TYPE_S32C1 = 0xc,  
    IVE_IMAGE_TYPE_U32C1 = 0xd,  
    IVE_IMAGE_TYPE_S64C1 = 0xe,  
    IVE_IMAGE_TYPE_U64C1 = 0xf,  
    IVE_IMAGE_TYPE_BF16C1 = 0x10,  
    IVE_IMAGE_TYPE_FP32C1 = 0x11,  
    IVE_IMAGE_TYPE_BUTT  
} IVE_IMAGE_TYPE_E;
```

## 【成员】

成员名称	描述
IVE_IMAGE_TYPE_U8C1	每个像素用 1 个 8bit 无符号数据表示的单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_S8C1	每个像素用 1 个 8bit 有符号数据表示的单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_YUV420SP	YUV420 Semiplanar 格式的图像。请参见图 1-3。
IVE_IMAGE_TYPE_YUV422SP	YUV422 Semiplanar 格式的图像。请参见图 1-4。
IVE_IMAGE_TYPE_YUV420P	YUV420 Planar 格式的图像。请参见图 1-5。
IVE_IMAGE_TYPE_YUV422P	YUV422 Planar 格式的图像。请参见图 1-6。
IVE_IMAGE_TYPE_S8C2_PACKAGE	每个像素用 2 个 8bit 有符号数据表示，且以 package 格式存储 2 通道图像。请参见图 1-7。
IVE_IMAGE_TYPE_S8C2_PLANAR	每个像素用 2 个 8bit 有符号数据表示，且以 planar 格式存储 2 通道图像。请参见图 1-8。
IVE_IMAGE_TYPE_S16C1	每个像素用 1 个 16bit 有符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_U16C1	每个像素用 1 个 16bit 无符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_U8C3_PACKAGE	每个像素用 3 个 8bit 无符号数据表示且以 Package 格式存储 3 通道图像。请参见图 1-9。
IVE_IMAGE_TYPE_U8C3_PLANAR	每个像素用 3 个 8bit 无符号数据表示 1 个像素的 3 通道图像，且以 planar 格式存储。请参见图 1-10。
IVE_IMAGE_TYPE_S32C1	每个像素用 1 个 32bit 有符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_U32C1	每个像素用 1 个 32bit 无符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_S64C1	每个像素用 1 个 64bit 有符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_U64C1	每个像素用 1 个 64bit 无符号数据表示单通道图像。请参见图 1-2。
IVE_IMAGE_TYPE_BF16C1	每个像素用 1 个 16bit 无符号数据表示单通道图像。
IVE_IMAGE_TYPE_UFP32C1	每个像素用 1 个 32bit 无符号数据表示单通道图像。

## 【注意事项】

无。

## 【相关数据类型及接口】

- IVE\_IMAGE\_S
- IVE\_SRC\_IMAGE\_S
- IVE\_DST\_IMAGE\_S

## 4.2.2 IVE\_IMAGE\_S

### 【说明】

定义二维广义图像信息。

### 【定义】

```
typedef struct _IVE_IMAGE_S
{
    IVE_IMAGE_TYPE_E enType;
    CVI_U64 u64phyAddr[3];
    CVI_U64 u64VirAddr[3];
    CVI_U32 u32Stride[3];
    CVI_U32 u32Width;
    CVI_U32 u32Height;
    CVI_U32 u32Reserved;
} IVE_IMAGE_S;
```

### 【成员】

成员名称	描述
enType	广义图像的图像类型。
U64phyAddr	广义图像的物理地址数组。
u64VirAddr	广义图像的虚拟地址数组。
u32Stride	广义图像的跨度。
u32Width	广义图像的宽度。
u32Height	广义图像的高度。
u32Reserved	保留位。

### 【注意事项】

无。

### 【相关数据类型及接口】

- IVE\_IMAGE\_TYPE\_E
- IVE\_SRC\_IMAGE\_S
- IVE\_DST\_IMAGE\_S

### 4.2.3 IVE\_SRC\_IMAGE\_S

**【说明】**

定义源图像。

**【定义】**

```
typedef IVE_IMAGE_S IVE_SRC_IMAGE_S;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

- IVE\_IMAGE\_S
- IVE\_DST\_IMAGE\_S

### 4.2.4 IVE\_DST\_IMAGE\_S

**【说明】**

定义输出图像。

**【定义】**

```
typedef IVE_IMAGE_S IVE_DST_IMAGE_S;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

- IVE\_IMAGE\_S
- IVE\_SRC\_IMAGE\_S

## 4.2.5 IVE\_DATA\_S

### 【说明】

定义以 byte 为单位的二维数据信息。

### 【定义】

```
typedef struct _IVE_DATA_S
{
    IVE_IMAGE_TYPE_E enType;

    CVI_U64 u64PhyAddr;

    CVI_U64 u64VirAddr;

    CVI_U32 u32Stride;

    CVI_U32 u32Width;

    CVI_U32 u32Height;

    CVI_U32 u32Reserved;
} IVE_DATA_S;
```

### 【成员】

成员名称	描述
u64PhyAddr	广义图像的物理地址数组。
u64VirAddr	广义图像的虚拟地址数组。
u32Stride	广义图像的跨度。
u32Width	广义图像的宽度。
u32Height	广义图像的高度。
u32Reserved	保留位。

### 【注意事项】

无。

### 【相关数据类型及接口】

无。



### 4.2.6 IVE\_SRC\_DATA\_S

**【说明】**

定义以 byte 为单位的二维源数据信息。

**【定义】**

```
typedef IVE_DATA_S IVE_SRC_DATA_S
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_IMAGE\_S

IVE\_DST\_DATA\_S

### 4.2.7 IVE\_DST\_DATA\_S

**【说明】**

定义 byte 为单位的二维输出数据信息。

**【定义】**

```
typedef IVE_DATA_S IVE_DST_DATA_S
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_IMAGE\_S

IVE\_SRC\_IMAGE\_S

## 4.2.8 IVE\_MEM\_INFO\_S

### 【说明】

定义一维数据内存信息。

### 【定义】

```
typedef struct _IVE_MEM_INFO_S
{
    CVI_U64 u64PhyAddr;

    CVI_U64 u64VirAddr;

    CVI_U32 u32Size;
} IVE_MEM_INFO_S;
```

### 【成员】

成员名称	描述
u64PhyAddr	一维数据物理地址。
u64VirAddr	一维数据虚拟地址。
u32Size	一维数据 byte 数目。

### 【注意事项】

无。

### 【相关数据类型及接口】

IVE\_SRC\_MEM\_INFO\_S

IVE\_DST\_MEM\_INFO\_S

## 4.2.9 IVE\_SRC\_MEM\_INFO\_S

### 【说明】

定义一维源数据。

### 【定义】

```
typedef IVE_MEM_INFO_S IVE_SRC_MEM_INFO_S;
```

### 【成员】

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_MEM\_INFO\_S

IVE\_DST\_MEM\_INFO\_S

## 4.2.10 IVE\_DST\_MEM\_INFO\_S

**【说明】**

定义一维源数据。

**【定义】**

```
typedef IVE_MEM_INFO_S IVE_DST_MEM_INFO_S;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_MEM\_INFO\_S

IVE\_SRC\_MEM\_INFO\_S

## 4.2.11 IVE\_8BIT\_U

**【说明】**

定义 8bit 数据联合体。

**【定义】**

```
typedef union _IVE_8BIT_U  
{  
    CVI_S8 s8Val;  
    CVI_U8 u8Val;  
} IVE_8BIT_U;
```

**【成员】**

成员名称	描述
s8Val	有符号 8bit 值。
u8Val	无符号 8bit 值。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.12 IVE\_POINT\_U16\_S

**【说明】**

定义 unsigned 16bit 坐标数据结构体。

**【定义】**

```
typedef struct _IVE_POINT_U16_S
{
    CVI_U16 u16X;
    CVI_U16 u16Y;
} IVE_POINT_U16_S;
```

**【成员】**

成员名称	描述
u16X	无符号 16bit X 坐标。
u16Y	无符号 16bit Y 坐标。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 4.2.13 IVE\_POINT\_S16\_S

**【说明】**

定义 signed 16bit 坐标数据结构体。

**【定义】**

```
typedef struct _IVE_POINT_S16_S
{
    CVI_S16 s16X;
    CVI_S16 s16Y;
} IVE_POINT_S16_S;
```

**【成员】**

成员名称	描述
s16X	有符号 16bit X 坐标。
s16Y	有符号 16bit Y 坐标。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### 4.2.14 IVE\_RECT\_U16\_S

**【说明】**

定义 unsigned 16bit 矩形数据结构体。

**【定义】**

```
typedef struct _IVE_RECT_U16_S
{
    CVI_U16 u16X;
    CVI_U16 u16Y;
    CVI_U16 u16Width;
```

(下页继续)

(续上页)

```
    CVI_U16 u16Height;  
} IVE_RECT_U16_S;
```

**【成员】**

成员名称	描述
u16X	有符号 16bit 矩形 X 坐标。
u16Y	有符号 16bit 矩形 Y 坐标。
u16Width	矩形宽度。
u16Height	矩形高度。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.15 IVE\_LOOK\_UP\_TABLE\_S

**【说明】**

定义 LUT 表格结构体。

**【定义】**

```
typedef struct _IVE_LOOK_UP_TABLE_S  
{  
    IVE_MEM_INFO_S stTable;  
    CVI_U16 u16ElemNum;  
    CVI_U8 u8TabInPreci;  
    CVI_U8 u8TabOutNorm;  
    CVI_S32 s32TabInLower;  
    CVI_S32 s32TabInUpper;  
} IVE_LOOK_UP_TABLE_S;
```

**【成员】**

成员名称	描述
stTable	LUT 表格内存。
u16ElemNum	LUT 表格元素数量。
u8TabInPreci	查找表原始输入的预定精度值。
u8TabOutNorm	查找表输出归一化参数。
s32TabInLower	查找表的原始输入下限。
s32TabInUpper	查找表的原始输入上限。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.16 IVE\_DMA\_MODE\_E

**【说明】**

定义 DMA 操作模式。

**【定义】**

```
typedef struct _IVE_DMA_MODE_E
{
    IVE_DMA_MODE_DIRECT_COPY = 0x0,
    IVE_DMA_MODE_INTERVAL_COPY = 0x1,
    IVE_DMA_MODE_SET_3BYTE = 0x2,
    IVE_DMA_MODE_SET_8BYTE = 0x3,
    IVE_DMA_MODE_BUTT
} IVE_DMA_MODE_E;
```

**【成员】**

成员名称	描述
IVE_DMA_MODE_DIRECT_COPY	直接快速拷贝模式。
IVE_DMA_MODE_INTERVAL_COPY	间隔拷贝模式, 请参见 CVI_IVE_DMA【注意】说明。
IVE_DMA_MODE_SET_3BYTE	3byte 赋值模式, 请参见 CVI_IVE_DMA【注意】说明。
IVE_DMA_MODE_SET_8BYTE	8byte 赋值模式, 请参见 CVI_IVE_DMA【注意】说明。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.17 IVE\_DMA\_CTRL\_S

**【说明】**

定义 DMA 控制信息。

**【定义】**

```
typedef struct _IVE_DMA_CTRL_S
{
    IVE_DMA_MODE_E enMode;

    CVI_U64 u64Val;

    CVI_U8 u8HorSegSize;

    CVI_U8 u8ElemSize;

    CVI_U8 u8VerSegRows;
}IVE_DMA_CTRL_S;
```

**【成员】**



成员名称	描述
enMode	DMA 操作模式。
u64Val	仅赋值模式使用，用于对内存赋值，3byte 赋值模式用低 3byte 保存。
u8HorSegSize	仅间隔拷贝模式使用，水平方向将源图像一行分割的段大小。取值范围:{2, 3, 4, 8, 16}。
u8ElemSize	仅间隔拷贝模式使用，分割的每一段中前 u8ElemSizebyte 为有效的拷贝字段。取值范围:[1, u8HorSegSize-1]。
u8VerSegRows	仅间隔拷贝模式使用，将每 u8VerSegRows 行中第一行数据分割为 u8HorSegSize 大小的段，拷贝每段中的前 u8ElemSize 大小的字节

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_DMA\_MODE\_E

## 4.2.18 IVE\_FILTER\_CTRL\_S

**【说明】**

定义模板滤波控制信息。

**【定义】**

```
typedef struct _IVE_FILTER_CTRL_S
{
    CVI_S8 as8Mask[25];

    CVI_U8 u8Norm;
} IVE_FILTER_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	5x5 模板系数，外围系数设为 0 可实现 3x3 模板滤波。
u8Norm	归一化参数。取值范围:[0, 13]。

**【注意事项】**

通过配置不同的模板系数可以达到不同的滤波效果。

**【相关数据类型及接口】**

无。

## 4.2.19 IVE\_CSC\_MODE\_E

**【说明】**

定义色彩空间转换模式。

**【定义】**

```
typedef enum _IVE_CSC_MODE_E
{
    IVE_CSC_MODE_VIDEO_BT601_YUV2RGB = 0x0,
    IVE_CSC_MODE_VIDEO_BT709_YUV2RGB = 0x1,
    IVE_CSC_MODE_PIC_BT601_YUV2RGB = 0x2,
    IVE_CSC_MODE_PIC_BT709_YUV2RGB = 0x3,
    IVE_CSC_MODE_PIC_BT601_YUV2HSV = 0x4,
    IVE_CSC_MODE_PIC_BT709_YUV2HSV = 0x5,
    IVE_CSC_MODE_PIC_BT601_YUV2LAB = 0x6,
    IVE_CSC_MODE_PIC_BT709_YUV2LAB = 0x7,
    IVE_CSC_MODE_VIDEO_BT601_RGB2YUV = 0x8,
    IVE_CSC_MODE_VIDEO_BT709_RGB2YUV = 0x9,
    IVE_CSC_MODE_PIC_BT601_RGB2YUV = 0xa,
    IVE_CSC_MODE_PIC_BT709_RGB2YUV = 0xb,
    IVE_CSC_MODE_BUTT
} IVE_CSC_MODE_E;
```

**【成员】**

成员名称	描述
IVE_CSC_MODE_VIDEO_BT601_YUV2RG	BT601 YUV2RGB Video 格式转换
IVE_CSC_MODE_VIDEO_BT709_YUV2RGB	BT709 YUV2RGB Video 格式转换
IVE_CSC_MODE_PIC_BT601_YUV2RGB	BT601 YUV2RGB 影像格式转换
IVE_CSC_MODE_PIC_BT709_YUV2RGB	BT709 YUV2RGB 影像格式转换
IVE_CSC_MODE_PIC_BT601_YUV2HSV	BT601 YUV2HSV 影像格式转换
IVE_CSC_MODE_PIC_BT709_YUV2HSV	BT709 YUV2HSV 影像格式转换
IVE_CSC_MODE_PIC_BT601_YUV2LAB	BT601 YUV 2LAB 影像格式转换
IVE_CSC_MODE_PIC_BT709_YUV2LAB	BT709 YUV 2LAB 影像格式转换
IVE_CSC_MODE_VIDEO_BT601_RGB2YUV	BT601 RGB2YUV Video 格式转换
IVE_CSC_MODE_VIDEO_BT709_RGB2YUV	BT709 RGB2YUV Video 格式转换
IVE_CSC_MODE_PIC_BT601_RGB2YUV	BT601 RGB2YUV 影像格式转换
IVE_CSC_MODE_PIC_BT709_RGB2YUV	BT709 RGB2YUV 影像格式转换

**【注意事项】****【相关数据类型及接口】**

IVE\_CSC\_CTRL\_S

## 4.2.20 IVE\_CSC\_CTRL\_S

**【说明】**

定义色彩空间转换控制信息。

**【定义】**

```
typedef struct _IVE_CSC_CTRL_S
{
    IVE_CSC_MODE_E.
    enMode;
}IVE_CSC_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	工作模式

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_CSC\_MODE\_E

## 4.2.21 IVE\_SOBEL\_OUT\_CTRL\_E

### 【说明】

定义 Sobel 输出控制信息。

### 【定义】

```
typedef enum _IVE_SOBEL_OUT_CTRL_E
{
    IVE_SOBEL_OUT_CTRL_BOTH = 0x0,
    IVE_SOBEL_OUT_CTRL_HOR = 0x1,
    IVE_SOBEL_OUT_CTRL_VER = 0x2,
    IVE_SOBEL_OUT_CTRL_BUTT
} IVE_SOBEL_OUT_CTRL_E;
```

### 【成员】

成员名称	描述
IVE_SOBEL_OUT_CTRL_BOTH	同时输出用模板和转置模板滤波的结果。
IVE_SOBEL_OUT_CTRL_HOR	仅输出用模板直接滤波的结果。
IVE_SOBEL_OUT_CTRL_VER	仅输出用转置模板滤波的结果。

### 【注意事项】

无。

### 【相关数据类型及接口】

IVE\_SOBEL\_CTRL\_S

## 4.2.22 IVE\_SOBEL\_CTRL\_S

### 【说明】

定义 Sobel-like 梯度计算控制信息。

### 【定义】

```
typedef struct _IVE_SOBEL_CTRL_S
{
```

(下页继续)

(续上页)

```
IVE_SOBEL_OUT_CTRL_E enOutCtrl;  
  
CVI_S8 as8Mask[25];  
} IVE_SOBEL_CTRL_S;
```

**【成员】**

成员名称	描述
enOutCtrl	输出控制枚举参数。
as8Mask[25]	模板系数。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_SOBEL\_OUT\_CTRL\_E

### 4.2.23 IVE\_MAG\_AND\_ANG\_OUT\_CTRL\_E

**【说明】**

定义梯度幅值与角度计算的输出格式。

**【定义】**

```
typedef struct _IVE_MAG_AND_ANG_OUT_CTRL_E  
{  
    IVE_MAG_AND_ANG_OUT_CTRL_MAG = 0x0,  
    IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG = 0x1,  
    IVE_MAG_AND_ANG_OUT_CTRL_BUTT  
} IVE_MAG_AND_ANG_OUT_CTRL_E;
```

**【成员】**

成员名称	描述
IVE_MAG_AND_ANG_OUT_CTRL_MAG	仅输出幅值
IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG	同时输出幅值和角度值

## 4.2.24 IVE\_MAG\_AND\_ANG\_CTRL\_S

### 【说明】

定义梯度幅值和幅角计算的控制信息。

### 【定义】

```
typedef struct _IVE_MAG_AND_ANG_CTRL_S
{
    IVE_MAG_AND_ANG_OUT_CTRL_E enOutCtrl;

    CVI_U16 u16Thr;

    CVI_S8 as8Mask[25];
} IVE_MAG_AND_ANG_OUT_CTRL_S;
```

### 【成员】

成员名称	描述
enOutCtrl	输出格式
u16Thr	阈值
as8Mask	5x5 Filter

### 【注意事项】

无。

### 【相关数据类型及接口】

IVE\_MAG\_AND\_ANG\_OUT\_CTRL\_E

## 4.2.25 IVE\_DILATE\_CTRL\_S

### 【说明】

定义膨胀控制信息。

### 【定义】

```
typedef struct _IVE_DILATE_CTRL_S
{
    CVI_U8 au8Mask[25];
} IVE_DILATE_CTRL_S;
```

**【成员】**

成员名称	描述
au8Mask[25]	5x5 模板系数。取值范围：0 或 255

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.26 IVE\_ERODE\_CTRL\_S

**【说明】**

定义腐蚀控制信息。

**【定义】**

```
typedef IVE_DILATE_CTRL_S IVE_ERODE_CTRL_S;
```

**【成员】**

成员名称	描述
au8Mask[25]	5x5 模板系数。取值范围：0 或 255

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.27 IVE\_THRESH\_MODE\_E

**【说明】**

定义图像二值化输出格式。

**【定义】**

```
typedef enum _IVE_THRESH_MODE_E  
{
```

(下页继续)

(续上页)

```

IVE_THRESH_MODE_BINARY = 0x0,
IVE_THRESH_MODE_TRUNC = 0x1,
IVE_THRESH_MODE_TO_MINVAL = 0x2,
IVE_THRESH_MODE_MIN_MID_MAX = 0x3,
IVE_THRESH_MODE_ORI_MID_MAX = 0x4,
IVE_THRESH_MODE_MIN_MID_ORI = 0x5,
IVE_THRESH_MODE_MIN_ORI_MAX = 0x6,
IVE_THRESH_MODE_ORI_MID_ORI = 0x7,
} IVE_THRESH_MODE_E;

```

**【成员】**

成员名称	描述
IVE_THRESH_MODE_BINARY	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{minVal}$ ; $\text{srcVal} > \text{lowThr}$ , $\text{dstVal} = \text{maxVal}$ 。
IVE_THRESH_MODE_TRUNC	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{srcVal}$ $\text{srcVal} > \text{lowThr}$ , $\text{dstVal} = \text{maxVal}$
IVE_THRESH_MODE_TO_MINVAL	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{minVal}$ $\text{srcVal} > \text{lowThr}$ , $\text{dstVal} = \text{srcVal}$
IVE_THRESH_MODE_MIN_MID_MAX	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{minVal}$ $\text{lowThr} < \text{srcVal} \leq \text{highThr}$ , $\text{dstVal} = \text{midVal}$ $\text{srcVal} > \text{highThr}$ , $\text{dstVal} = \text{maxVal}$
IVE_THRESH_MODE_ORI_MID_MAX	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{srcVal}$ $\text{lowThr} < \text{srcVal} \leq \text{highThr}$ $\text{dstVal} = \text{midVal}$ $\text{srcVal} > \text{highThr}$ , $\text{dstVal} = \text{maxVal}$
IVE_THRESH_MODE_MIN_MID_ORI	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{minVal}$ $\text{lowThr} < \text{srcVal} \leq \text{highThr}$ $\text{dstVal} = \text{midVal}$ $\text{srcVal} > \text{highThr}$ , $\text{dstVal} = \text{srcVal}$
IVE_THRESH_MODE_MIN_ORI_MAX	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{minVal}$ $\text{lowThr} < \text{srcVal} \leq \text{highThr}$ $\text{dstVal} = \text{srcVal}$ $\text{srcVal} > \text{highThr}$ , $\text{dstVal} = \text{maxVal}$
IVE_THRESH_MODE_ORI_MID_ORI	$\text{srcVal} \leq \text{lowThr}$ , $\text{dstVal} = \text{srcVal}$ $\text{lowThr} < \text{srcVal} \leq \text{highThr}$ $\text{dstVal} = \text{midVal}$ $\text{srcVal} > \text{highThr}$ , $\text{dstVal} = \text{srcVal}$

**【注意事项】**

无。

**【相关数据类型及接口】**



IVE\_THRESH\_CTRL\_S

## 4.2.28 IVE\_THRESH\_CTRL\_S

### 【说明】

定义图像二值化控制信息。

### 【定义】

```
typedef struct _IVE_THRESH_CTRL_S
{
    IVE_THRESH_MODE_E enMode;

    CVI_U8 u8LowThr;

    CVI_U8 u8HighThr;

    CVI_U8 u8MinVal;

    CVI_U8 u8MidVal;

    CVI_U8 u8MaxVal;
}IVE_THRESH_CTRL_S;
```

### 【成员】

成员名称	描述
enMode	阈值化运算模式。
u8LowThr	低阈值。取值范围：[0,255]。
u8HighThr	高阈值。取值范围：[0,255]。
u8MinVal	最小值。取值范围：[0,255]。
u8MidVal	中值。取值范围：[0,255]。
u8MaxVal	最大值。取值范围：[0,255]。

### 【注意事项】

无。

### 【相关数据类型及接口】

IVE\_THRESH\_MODE\_E

### 4.2.29 IVE\_SUB\_MODE\_E

**【说明】**

定义两图像相减输出格式。

**【定义】**

```
Typedef enum _IVE_SUB_MODE_E
{
    IVE_SUB_MODE_ABS = 0x0,
    IVE_SUB_MODE_SHIFT = 0x1,
    IVE_SUB_MODE_BUTT
} IVE_SUB_MODE_E;
```

**【成员】**

成员名称	描述
IVE_SUB_MODE_ABS	相减取绝对值。
IVE_SUB_MODE_SHIFT	将结果右移一位输出，保留符号位。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_SUB\_CTRL\_S

### 4.2.30 IVE\_SUB\_CTRL\_S

**【说明】**

定义两图像相减控制参数。

**【定义】**

```
Typedef struct IVE_SUB_CTRL
{
    IVE_SUB_MODE_E enMode;
} IVE_SUB_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	两图像相减模式

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_SUB\_MODE\_E

### 4.2.31 IVE\_INTEG\_OUT\_CTRL\_E

**【说明】**

定义积分图输出控制参数。

**【定义】**

```
Typedef enum _IVE_INTEG_OUT_CTRL_E
{
    IVE_INTEG_OUT_CTRL_COMBINE = 0x0,
    IVE_INTEG_OUT_CTRL_SUM = 0x1,
    IVE_INTEG_OUT_CTRL_SQSUM = 0x2,
    IVE_INTEG_OUT_CTRL_BUTT
} IVE_INTEG_OUT_CTRL_E;
```

**【成员】**

成员名称	描述
IVE_INTEG_OUT_CTRL_COMBINE	和、平方和积分图组合输出，
IVE_INTEG_OUT_CTRL_SUM	仅和积分图输出。
IVE_INTEG_OUT_CTRL_SQSUM	仅平方和积分图输出。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_INTEG\_CTRL\_S

4.2.32 IVE\_INTEG\_CTRL\_S

【说明】

定义积分图计算控制参数。

【定义】

```
Typedef struct _IVE_INTEG_CTRL_S
{
    IVE_INTEG_MODE_E enOutCtrl;
} IVE_INTEG_CTRL_S;
```

【成员】

成员名称	描述
enOutCtrl	积分图输出控制参数

【注意事项】

无。

【相关数据类型及接口】

IVE\_INTEG\_OUT\_CTRL\_E

4.2.33 IVE\_THRESH\_S16\_MODE\_E

【说明】

定义 16bit 有符号图像的阈值化模式。

【定义】

```
typedef enum _IVE_THRESH_S16_MODE_E
{
    IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX = 0x0,
    IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX = 0x1,
    IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX = 0x2,
    IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX = 0x3,
    IVE_INTEG_MODE_E enOutCtrl;
```

(下页继续)

(续上页)

```
} IVE_THRESH_S16_MODE_E;
```

**【成员】**

成员名称	描述
IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX	srcVal $\leq$ lowThr, dstVal = minVal; lowThr < srcVal $\leq$ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX	srcVal $\leq$ lowThr, dstVal = minVal; lowThr < srcVal $\leq$ highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX	srcVal $\leq$ lowThr, dstVal = minVal; lowThr < srcVal $\leq$ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX	srcVal $\leq$ lowThr, dstVal = minVal; lowThr < srcVal $\leq$ highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal;

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_THRESH\_S16\_CTRL\_S

### 4.2.34 IVE\_THRESH\_S16\_CTRL\_S

**【说明】**

定义 16bit 有符号图像的阈值化控制参数。

**【定义】**

```
typedef struct _IVE_THRESH_S16_CTRL_S
{
    IVE_THRESH_S16_MODE_E enMode;

    CVI_S16 s16LowThr;

    CVI_S16 s16HightThr;

    IVE_8BIT_U un8MinVal;

    IVE_8BIT_U un8MidVal;

    IVE_8BIT_U un8MaxVal;
} IVE_THRESH_S16_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	阈值化运算模式。
s16LowThr	低阈值。
s16HightThr	高阈值。
un8MinVal	最小值。
un8MidVal	中间值。
un8MaxVal	最大值。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_THRESH\_S16\_MODE\_E

### 4.2.35 IVE\_THRESH\_U16\_MODE\_E

**【说明】**

定义 16bit 无符号图像的阈值化控制参数。

**【定义】**

```
typedef struct _IVE_THRESH_U16_MODE_E
{
    IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX=0x0,
    IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX=0x1,
    IVE_THRESH_U16_MODE_BUTT
} IVE_THRESH_U16_MODE_E;
```

**【成员】**

成员名称	描述
IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal;

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_THRESH\_U16\_CTRL\_S

### 4.2.36 IVE\_THRESH\_U16\_CTRL\_S

**【说明】**

定义 16bit 无符号图像的阈值化控制参数。

**【定义】**

```
typedef struct _IVE_THRESH_U16_CTRL_S
{
    IVE_THRESH_U16_MODE_E enMode;

    CVI_U16 u16LowThr;

    CVI_U16 u16HightThr;

    IVE_8BIT_U u8MinVal;

    IVE_8BIT_U u8MidVal;

    IVE_8BIT_U u8MaxVal;
} IVE_THRESH_U16_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	阈值化运算模式。
u16LowThr	低阈值。
u16HightThr	高阈值。
u8MinVal	最小值。
u8MidVal	中间值。
u8MaxVal	最大值。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_THRESH\_S16\_MODE\_E



### 4.2.37 IVE\_16BIT\_TO\_8BIT\_MODE\_E

**【说明】**

定义 16bit 图像数据到 8bit 图像数据的转化模式

**【定义】**

```
typedef enum _IVE_16BIT_TO_8BIT_MODE_E
{
    IVE_16BIT_TO_8BIT_MODE_S16_TO_S8=0x0,
    IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS=0x1,
    IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS=0x2,
    IVE_16BIT_TO_8BIT_MODE_S16_TO_U8=0x3,
    IVE_16BIT_TO_8BIT_MODE_BUTT
} IVE_16BIT_TO_8BIT_MODE_E;
```

**【成员】**

成员名称	描述
IVE_16BIT_TO_8BIT_MODE_S16_TO_S8	S16 数据到 S8 数据的线性换。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS	S16 数据线性变换到 S8 数据后取绝对值得到 S8 数据。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS	S16 数据线性变换到 S8 数据且平移后截断到 U8 数据。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8	U16 数据线性变换到 U8 数据。

**【注意事项】**

无。

**【相关数据类型及接口】**

- IVE\_16BIT\_TO\_8BIT\_CTRL\_S

### 4.2.38 IVE\_16BIT\_TO\_8BIT\_CTRL\_S

**【说明】**

定义 16bit 图像数据到 8bit 图像数据的转化控制参数

**【定义】**

```
typedef struct _IVE_16BIT_TO_8BIT_CTRL_S
{
    IVE_16BIT_TO_8BIT_MODE_E enMode;

    CVI_U16 u16Denominator;

    CVI_U8 u8Numerator;

    CVI_S8 s8Bias;
} IVE_16BIT_TO_8BIT_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	16bit 数据到 8bit 数据的转换模式。
u16Denominator	线性变换中的分母。取值范围： $[\max\{1, u8Numerator\}, 65535]$
u8Numerator	线性变换中的分子。取值范围： $[0, 255]$ 。
s8Bias	线性变换中的平移项。取值范围： $[-128, 127]$ 。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_16BIT\_TO\_8BIT\_MODE\_E

### 4.2.39 IVE\_ORD\_STAT\_FILTER\_MODE\_E

**【说明】**

定义顺序统计量滤波模式

**【定义】**

```
typedef enum IVE_ORD_STAT_FILTER_MODE
```

(下页继续)

(续上页)

```
{  
    IVE_ORD_STAT_FILTER_MODE_MEDIAN = 0x0,  
    IVE_ORD_STAT_FILTER_MODE_MIN = 0x1,  
    IVE_ORD_STAT_FILTER_MODE_MAX = 0x2,  
    IVE_ORD_STAT_FILTER_MODE_BUTT  
} IVE_ORD_STAT_FILTER_MODE_E;
```

**【成员】**

成员名称	描述
IVE_ORD_STAT_FILTER_MODE_MEDIAN	中值滤波
IVE_ORD_STAT_FILTER_MODE_MIN	最小值滤波，等价于灰度图的腐蚀。
IVE_ORD_STAT_FILTER_MODE_MAX	最大值滤波，等价于灰度图的膨胀。

**【注意事项】**

无。

**【相关数据类型及接口】**

· ORD\_STAT\_FILTER\_CTRL\_S

## 4.2.40 IVE\_ORD\_STAT\_FILTER\_CTRL\_S

**【说明】**

定义顺序统计量滤波控制参数

**【定义】**

```
typedef struct _IVE_ORD_STAT_FILTER_CTRL_S  
{  
    IVE_ORD_STAT_FILTER_MODE_E enMode;  
} IVE_ORD_STAT_FILTER_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	顺序统计量滤波模式

**【注意事项】**

无。

**【相关数据类型及接口】**

· IVE\_ORD\_STAT\_FILTER\_MODE\_E

## 4.2.41 IVE\_EQUALIZE\_HIST\_CTRL\_S

**【说明】**

定义直方图均衡化控制参数

**【定义】**

```
typedef struct _IVE_EQUALIZE_HIST_CTRL_S
{
    IVE_MEM_INFO_S stMem;
} IVE_EQUALIZE_HIST_CTRL_S;
```

**【成员】**

成员名称	描述
stMem	直方图均衡化控制参数

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.42 IVE\_MAP\_MODE\_E

**【说明】**

MAP 模式。

**【定义】**

```
typedef enum _IVE_MAP_MODE_E
{
```

(下页继续)

(续上页)

```
IVE_MAP_MODE_U8 = 0x0;

IVE_MAP_MODE_S16 = 0x1;

IVE_MAP_MODE_U16 = 0x2;

} IVE_MAP_MODE_E;
```

**【成员】**

成员名称	描述
IVE_MAP_MODE_U8	U8C1 到 U8C1Mapping
IVE_MAP_MODE_S16	U8C1 到 U16C1Mapping
IVE_MAP_MODE_U16	U8C1 到 S16C1Mapping

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_MAP\_CTRL\_S

### 4.2.43 IVE\_MAP\_CTRL\_S

**【说明】**

定义俩图像的加权加控制参数。

**【定义】**

```
typedef struct _IVE_MAP_CTRL_S
{
    IVE_MAP_MODE_E enMode;
} IVE_MAP_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	MAP 的模式。

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

#### 4.2.44 IVE\_MAP\_U8BIT\_LUT\_MEM\_S

##### 【说明】

该结构体用于存储 U8 位查找表的内存映射数据。结构体中包含一个定长数组，用于存储查找表中的数据，每个数组元素对应 LUT 的一个条目。

##### 【定义】

```
typedef struct _IVE_MAP_U8BIT_LUT_MEM_S {  
  
    CVI_U8 au8Map[IVE_MAP_NUM];  
  
} IVE_MAP_U8BIT_LUT_MEM_S;
```

##### 【成员】

成员名称	描述
au8Map	数组长度由宏 IVE_MAP_NUM 定义，用于存储查找表数据。

##### 【注意事项】

IVE\_MAP\_NUM = 256

##### 【相关数据类型】

无。

#### 4.2.45 IVE\_MAP\_U16BIT\_LUT\_MEM\_S

##### 【说明】

该结构体用于存储无符号 16 位查找表的内存映射数据。结构体中包含一个定长数组，用于存储查找表中的数据，每个数组元素对应 LUT 的一个条目。

##### 【定义】

```
typedef struct _IVE_MAP_U16BIT_LUT_MEM_S {  
  
    CVI_U16 au16Map[IVE_MAP_NUM];  
  
} IVE_MAP_U16BIT_LUT_MEM_S;
```

##### 【成员】

成员名称	描述
au16Map	数组长度由宏 IVE_MAP_NUM 定义，用于存储查找表数据。

**【注意事项】**

IVE\_MAP\_NUM = 256

**【相关数据类型】**

无。

## 4.2.46 IVE\_MAP\_S16BIT\_LUT\_MEM\_S

**【说明】**

该结构体用于存储有符号 16 位查找表的内存映射数据。结构体中包含一个定长数组，用于存储查找表中的数据，每个数组元素对应 LUT 的一个条目

**【定义】**

```
typedef struct _IVE_MAP_S16BIT_LUT_MEM_S {  
    CVI_S16 as16Map[IVE_MAP_NUM];  
} IVE_MAP_S16BIT_LUT_MEM_S;
```

**【成员】**

成员名称	描述
as16Map	数组长度由宏 IVE_MAP_NUM 定义，用于存储查找表数据

**【注意事项】**

IVE\_MAP\_NUM = 256

**【相关数据类型】**

无。

## 4.2.47 IVE\_EQUALIZE\_HIST\_CTRL\_MEM\_S

**【说明】**

定义用于直方图均衡化控制内存，包括保存直方图统计数据与均衡化映射的数组。

**【定义】**

```
typedef struct _IVE_EQUALIZE_HIST_CTRL_MEM_S  
{  
    CVI_U32 au32Hist[IVE_HIST_NUM];  
    CVI_U8 au8Map[IVE_MAP_NUM];  
} IVE_EQUALIZE_HIST_CTRL_MEM_S;
```

## 【成员】

成员名称	描述
au32Hist	用于存储直方图数据的数组，其大小由宏 IVE_HIST_NUM 定义。数组中各元素表示直方图不同区间的统计数据。
au8Map	用于存储直方图均衡化映射的数组，其大小由宏 IVE_MAP_NUM 定义。数据用于映射输入灰度值到均衡化后的灰度值。

## 【注意事项】

无。

## 【相关数据类型及接口】

IVE\_MAP\_MODE\_E

## 4.2.48 IVE\_CANNY\_HYS\_EDGE\_CTRL\_S

## 【说明】

该结构体用于存储 Canny 边缘检测算法中滤波和阈值控制的信息。结构体包含内存信息、低高阈值以及用于边缘检测的掩码数组，确保算法在边缘检测过程中有效过滤和判断边缘强度。

## 【定义】

```
typedef struct _IVE_CANNY_HYS_EDGE_CTRL_S {  
  
    IVE_MEM_INFO_S stMem;  
  
    CVI_U16 u16LowThr;  
  
    CVI_U16 u16HighThr;  
  
    CVI_S8 as8Mask[25];  
  
} IVE_CANNY_HYS_EDGE_CTRL_S;
```

## 【成员】

成员名称	描述
stMem	内存信息结构体，用于描述与边缘检测相关的内存分配或布局信息。
u16LowThr	表示边缘检测的低阈值，用于控制滤波器在检测较弱边缘时的灵敏度。
u16HighThr	表示边缘检测的高阈值，用于控制滤波器在检测显著边缘时的灵敏度。
as8Mask	用作 Canny 边缘检测中的掩码，用于进一步处理边缘检测算法的输出。



**【注意事项】**

无。

**【相关数据类型】**

无。

## 4.2.49 IVE\_CANNY\_STACK\_SIZE\_S

**【说明】**

该结构体用于存储 Canny 算法输出的堆栈大小相关信息。结构体包含输出堆栈大小和对齐需要的保留空间，确保内存对齐符合要求。

**【定义】**

```
typedef struct _IVE_CANNY_STACK_SIZE_S {  
  
    CVI_U32 u32StackSize;  
  
    CVI_U8 u8Reserved[12];  
  
} IVE_CANNY_STACK_SIZE_S;
```

**【成员】**

成员名称	描述
u32StackSize	表示 Canny 算法输出的堆栈大小，用于分配足够的内存空间以存储处理结果。
u8Reserved	为了满足 16 字节对齐的要求而预留的空间。

**【注意事项】**

无。

**【相关数据类型】**

无。

## 4.2.50 IVE\_ADD\_CTRL\_S

**【说明】**

定义俩图像的加权加控制参数。

**【定义】**

```
typedef struct _IVE_ADD_CTRL_S
{
    CVI_U0Q16 u0q16X;

    CVI_U0Q16 u0q16Y;
} IVE_ADD_CTRL_S;
```

**【成员】**

成员名称	描述
u0q16X	加权加 “xA+yB” 中的权重 “x”
u0q16Y	加权加 “xA+yB” 中的权重 “y”

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.51 IVE\_NCC\_DST\_MEM\_S

**【说明】**

定义 NCC 的输出内存信息。

**【定义】**

```
typedef struct _IVE_NCC_DST_MEM_S
{
    CVI_U64 u64Numerator;

    CVI_U64 u64QuadSum1;

    CVI_U64 u64QuadSum2;

    CVI_U8 u8Reserved[8];
} IVE_NCC_DST_MEM_S;
```

**【成员】**

成员名称	描述
u64Numerator	NCC 计算公式的分子-- $\sum_{i=1}^w \sum_{j=1}^h (I_{src1}(i, j) * I_{src2}(i, j))$
u64QuadSum1	NCC 计算公式的分母--根号内部分： $\sum_{i=1}^w \sum_{j=1}^h (I_{src1}^2(i, j))。$
u64QuadSum2	NCC 计算公式的分母--根号内部分： $\sum_{i=1}^w \sum_{j=1}^h (I_{src2}^2(i, j))$
u8Reserved	保留字段。

**【注意事项】**

计算公式参考 CVI\_IVE\_NCC 中的 [注意]。

**【相关数据类型及接口】**

无。

## 4.2.52 IVE\_REGION\_S

**【说明】**

定义连通区域信息。

**【定义】**

```
typedef struct _IVE_REGION_S {  
  
    CVI_U32 u32Area;  
  
    CVI_U16 u16Left;  
  
    CVI_U16 u16Right;  
  
    CVI_U16 u16Top;  
  
    CVI_U16 u16Bottom;  
  
} IVE_REGION_S;
```

**【成员】**

成员名称	描述
u32Area	连通区域面积，以连通区域像素数目表示。
u16Left	连通区域外接矩形的最左边坐标。
u16Right	连通区域外接矩形的最右边坐标。
u16Top	连通区域外接矩形的最上面坐标。
u16Bottom	连通区域外接矩形的最下面坐标。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_CCBLOB\_S

### 4.2.53 IVE\_CCBLOB\_S

**【说明】**

定义连通区域标记的输出信息。

**【定义】**

```
typedef struct _IVE_CCBLOB_S {  
  
    CVI_U16 u16CurAreaThr;  
  
    CVI_S8 s8LabelStatus;  
  
    CVI_U8 u8RegionNum;  
  
    IVE_REGION_S astRegion[IVE_MAX_REGION_NUM];  
} IVE_CCBLOB_S;
```

**【成员】**

成员名称	描述
u16CurAreaThr	有效连通区域的面积阈值，astRegion 中面积小于这个阈值的都被置为 0。
s8LabelStatus	连通区域标记是否成功。 -1：标记失败； 0：标记成功。
u8RegionNum	有效连通区域个数。
astRegion[IVE_MAX_REGION_NUM]	连通区域信息：有效的连通区域其面积大于 0，对应标记为数组下标加 1。

**【注意事项】**

无。

【相关数据类型及接口】

IVE\_REGION\_S

## 4.2.54 IVE\_CCL\_MODE\_E

【说明】

定义连通区域模式。

【定义】

```
typedef enum _IVE_CCL_MODE_E {  
  
    IVE_CCL_MODE_4C = 0x0,  
  
    IVE_CCL_MODE_8C = 0x1,  
  
    IVE_CCL_MODE_BUTT  
}IVE_CCL_MODE_E;
```

【成员】

成员名称	描述
IVE_CCL_MODE_4C	4 连通
IVE_CCL_MODE_8C	8 连通

【注意事项】

无。

【相关数据类型及接口】

无。

## 4.2.55 IVE\_CCL\_CTRL\_S

【说明】

定义连通区域标记控制参数。

【定义】

```
typedef struct _IVE_CCL_CTRL_S {  
  
    IVE_CCL_MODE_E enMode;
```

(下页继续)

(续上页)

```
    CVI_U16 u16InitAreaThr;

    CVI_U16 u16Step;
} IVE_CCL_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	连通区域模式。
u16InitAreaThr	初始面积阈值。 取值范围：[0x0, 0xFFFF]。 参考取值：4。
u16Step	面积阈值增长步长。 取值范围：[0x1, 0xFFFF]。 参考取值：2。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_CCBLOB\_S

## 4.2.56 IVE\_GMM\_CTRL\_S

**【说明】**

定义 GMM 控制参数。

**【定义】**

```
typedef struct _IVE_GMM_CTRL_S {

    CVI_U22Q10 u22q10NoiseVar;

    CVI_U22Q10 u22q10MaxVar;

    CVI_U22Q10 u22q10MinVar;

    CVI_U0Q16 u0q16LearnRate;

    CVI_U0Q16 u0q16BgRatio;

    CVI_U8Q8 u8q8VarThr;
```

(下页继续)

(续上页)

```

CVI_U0Q16 u0q16InitWeight;

CVI_U8 u8ModelNum;

} IVE_GMM_CTRL_S;

```

**【成员】**

成员名称	描述
u22q10NoiseVar	初始噪声方差数值范围：[0x1, 0xFFFFFFFF]
u22q10MaxVar	模型最大方差数值范围：[0x1, 0xFFFFFFFF]
u22q10MinVar	模型最小方差数值范围：[1, u22q10MaxVar]
u0q16LearnRate	学习率数值范围：[1, 65535]
u0q16BgRatio	背景比率阈值数值范围：[1, 65535]
u8q8VarThr	变异数阈值数值范围：[1, 65535]
u0q16InitWeight	初始权重数值范围：[1, 65535]
u8ModelNum	几个 Gaussian 模型数值范围：{3, 5}

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

## 4.2.57 IVE\_LBP\_CMP\_MODE\_E

**【说明】**

定义 LBP 计算的比较模式。

**【定义】**

```

typedef enum _IVE_LBP_CMP_MODE_E
{
    IVE_LBP_CMP_MODE_NORMAL = 0x0,
    IVE_LBP_CMP_MODE_ABS = 0x1,
    IVE_LBP_CMP_MODE_BUTT
} IVE_LBP_CMP_MODE_E;

```

**【成员】**

成员名称	描述
IVE_LBP_CMP_MODE_NORMAL	LBP 简单比较模式
IVE_LBP_CMP_MODE_ABS	LBP 绝对值比较模式

**【注意事项】**

计算公式参考 CVI\_IVE\_LBP 中的 [注意]。

**【相关数据类型及接口】**

IVE\_LBP\_CTRL\_S。

## 4.2.58 IVE\_LBP\_CTRL\_S

**【说明】**

定义 LBP 纹理计算控制参数。

**【定义】**

```
Typedef struct _IVE_LBP_CTRL_S
{
    IVE_LBP_CMP_MODE_E enMode;

    IVE_8BIT_U un8BitThr;
}IVE_LBP_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	LBP 比较模式
un8BitThr	LBP 比较阈值。 IVE_LBP_CMP_MODE_NORMAL 下的取值范围：[-128,127] IVE_LBP_CMP_MODE_ABS 下的取值范围：[0,255]

**【注意事项】**

计算公式参考 CVI\_IVE\_LBP 中的 [注意]。

**【相关数据类型及接口】**



IVE\_LBP\_CMP\_MODE\_E

IVE\_8BIT\_U

### 4.2.59 IVE\_NORM\_GRAD\_OUT\_CTRL\_E

#### 【说明】

定义归一化梯度信息计算任务输出控制枚举类型。

#### 【定义】

```
typedef enum _IVE_NORM_GRAD_OUT_CTRL_E
{
    IVE_NORM_GRAD_OUT_CTRL_HOR_AND_VER = 0x0,
    IVE_NORM_GRAD_OUT_CTRL_HOR = 0x1,
    IVE_NORM_GRAD_OUT_CTRL_VER = 0x2,
    IVE_NORM_GRAD_OUT_CTRL_COMBINE = 0x3,
    IVE_NORM_GRAD_OUT_CTRL_BUTT
} IVE_NORM_GRAD_CTRL_E;
```

#### 【成员】

成员名称	描述
IVE_NORM_GRAD_OUT_CTRL_HOR_AND_VER	同时输出梯度信息的 H、V 分量图
IVE_NORM_GRAD_OUT_CTRL_HOR	仅输出梯度信息的 H 分量图。
IVE_NORM_GRAD_OUT_CTRL_VER	仅输出梯度信息的 V 分量图。
IVE_NORM_GRAD_OUT_CTRL_COMBINE	输出梯度信息以 package 存储

#### 【注意事项】

无。

#### 【相关数据类型及接口】

IVE\_NORM\_GRAD\_OUT\_CTRL\_S

## 4.2.60 IVE\_NORM\_GRAD\_CTRL\_S

### 【说明】

定义归一化梯度信息计算控制参数

### 【定义】

```
typedef struct _IVE_NORM_GRAD_CTRL_S {  
    IVE_NORM_GRAD_OUT_CTRL_E enOutCtrl;  
  
    CVI_S8 as8Mask[25];  
  
    CVI_U8 u8Norm;  
} IVE_NORM_GRAD_CTRL_S;
```

### 【成员】

成员名称	描述
enOutCtrl	输出格式
as8Mask[25]	计算梯度需要的模板
u8Norm	归一化参数

### 【注意事项】

无。

### 【相关数据类型及接口】

IVE\_NORM\_GRAD\_OUT\_CTRL\_E

## 4.2.61 IVE\_ST\_CANDI\_CORNER\_CTRL\_S

### 【说明】

该结构体用于存储角点检测相关的配置信息。结构体包含用于角点检测的内存信息和质量评价参数，以便在角点检测过程中控制检测的质量水平。

### 【定义】

```
typedef struct _IVE_ST_CANDI_CORNER_CTRL_S {  
  
    IVE_MEM_INFO_S stMem;  
  
    CVI_U0Q8 u0q8QualityLevel;  
} IVE_ST_CANDI_CORNER_CTRL_S;
```

**【成员】**

成员名称	描述
stMem	包含用于角点检测的内存信息，用于描述相关的内存分配或布局情况。
u0q8QualityLevel	表示角点检测时的质量等级，用于调整检测参数以获得期望的检测效果。

**【注意事项】**

无。

**【相关数据类型】**

无。

## 4.2.62 IVE\_ST\_CORNER\_INFO\_S

**【说明】**

该结构体用于存储角点检测结果的信息。结构体中包含检测到的角点数量以及一个数组，其中存放每个检测到的角点的坐标信息。

**【定义】**

```
typedef struct _IVE_ST_CORNER_INFO_S {  
    CVI_U16 u16CornerNum;  
    IVE_POINT_U16_S astCorner[IVE_ST_MAX_CORNER_NUM];  
} IVE_ST_CORNER_INFO_S;
```

**【成员】**

成员名称	描述
u16CornerNum	表示检测到的角点数目，用于指示有效角点在数组中的数量。
astCorner	其长度为 IVE_ST_MAX_CORNER_NUM, 用于存放每个检测到的角点的坐标信息。

**【注意事项】**

IVE\_ST\_MAX\_CORNER\_NUM = 500

**【相关数据类型】**

IVE\_POINT\_U16\_S

### 4.2.63 IVE\_ST\_CORNER\_CTRL\_S

**【说明】**

该结构体用于控制角点检测的参数设置，主要包括角点检测的最大数量和角点之间的最小距离。这些参数可用于优化角点检测算法的性能与效果。

**【定义】**

```
typedef struct _IVE_ST_CORNER_CTRL_S {  
    CVI_U16 u16MaxCornerNum;  
    CVI_U16 u16MinDist;  
} IVE_ST_CORNER_CTRL_S;
```

**【成员】**

成员名称	描述
u16MaxCornerNum	表示期望检测到的角点的最大数量，用于限制角点检测结果中角点的总数。
u16MinDist	表示检测到的角点之间的最小距离，以避免过于接近的角点被重复检测。

**【注意事项】**

无。

**【相关数据类型】**

无

### 4.2.64 IVE\_GRAD\_FG\_MODE\_E

**【说明】**

该枚举类型用于定义梯度前景模式，指示当前在梯度前景算法中所采用的处理策略。不同的枚举值代表不同的梯度处理方式。

**【定义】**

```
typedef enum _IVE_GRAD_FG_MODE_E {  
    IVE_GRAD_FG_MODE_USE_CUR_GRAD = 0x0,  
    IVE_GRAD_FG_MODE_FIND_MIN_GRAD = 0x1,  
    IVE_GRAD_FG_MODE_BUTT  
} IVE_GRAD_FG_MODE_E;
```

【成员】

成员名称	描述
IVE_GRAD_FG_MODEL	表示使用当前梯度作为前景的判断依据。
IVE_GRAD_FG_MODEL_MIN	表示在检测过程中查找最小梯度值以确定前景信息。
IVE_GRAD_FG_MODEL_BOUND	枚举边界值，用于标识当前枚举类型的结束或无效值，通常不用于实际计算。

【注意事项】

无。

【相关数据类型】

无。

4.2.65 IVE\_FG\_STAT\_DATA\_S

【说明】

该结构体用于保存前景统计数据，主要包含像素数量、亮度总和，以及为未来扩展预留的字段。此数据通常用于分析图像中的前景区域特性。

【定义】

```
typedef struct _IVE_FG_STAT_DATA_S {  
  
    CVI_U32 u32PixNum;  
  
    CVI_U32 u32SumLum;  
  
    CVI_U8 u8Reserved[8];  
  
} IVE_FG_STAT_DATA_S;
```

【成员】

成员名称	描述
u32PixNum	表示前景区域的像素数量，用于衡量该区域的大小。
u32SumLum	表示前景区域内所有像素亮度值的总和，可用于计算平均亮度。
u8Reserved	用于未来扩展使用，当前未作具体定义。

【注意事项】

无。

【相关数据类型】

无。

## 4.2.66 IVE\_BG\_STAT\_DATA\_S

### 【说明】

该结构体用于保存背景统计数据，主要包含像素数量、亮度总和，以及为未来扩展预留的字段。此数据通常用于分析图像背景区域的特性。

### 【定义】

```
typedef struct _IVE_BG_STAT_DATA_S {  
  
    CVI_U32 u32PixNum;  
  
    CVI_U32 u32SumLum;  
  
    CVI_U8 u8Reserved[8];  
  
} IVE_BG_STAT_DATA_S;
```

### 【成员】

成员名称	描述
u32PixNum	表示背景区域的像素数量，用于衡量该区域的面积大小。
u32SumLum	表示背景区域内所有像素亮度值的总和，可用于后续计算背景亮度均值。
u8Reserved	用于未来扩展使用，目前未具体定义。

### 【注意事项】

无。

### 【相关数据类型】

无。

## 4.2.67 IVE\_SAD\_MODE\_E

### 【说明】

定义 SAD 计算模式。

### 【定义】

```
typedef enum _IVE_SAD_MODE_E  
{  
  
    IVE_SAD_MODE_MB_4x4 = 0x0,
```

(下页继续)

(续上页)

```
IVE_SAD_MODE_MB_8x8 = 0x1,  
  
IVE_SAD_MODE_MB_16x16 = 0x2,  
  
IVE_NORM_GRAD_OUT_CTRL_BUTT  
} IVE_SAD_MODE_E;
```

**【成员】**

成员名称	描述
IVE_SAD_MODE_MB_4x4	按 4x4 像素块计算 SAD。
IVE_SAD_MODE_MB_8x8	按 8x8 像素块计算 SAD。
IVE_SAD_MODE_MB_16x16	按 16x16 像素块计算 SAD。

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_SAD\_CTRL\_S

## 4.2.68 IVE\_SAD\_OUT\_CTRL\_E

**【说明】**

定义 SAD 计算模式。

**【定义】**

```
typedef enum _IVE_SAD_OUT_CTRL_E  
{  
  
    IVE_SAD_OUT_CTRL_16BIT_BOTH = 0x0,  
  
    IVE_SAD_OUT_CTRL_8BIT_BOTH = 0x1,  
  
    IVE_SAD_OUT_CTRL_16BIT_SAD = 0x2,  
  
    IVE_SAD_OUT_CTRL_8BIT_SAD = 0x3,  
  
    IVE_SAD_OUT_CTRL_THRESH = 0x4,  
  
    IVE_SAD_OUT_CTRL_BUTT  
} IVE_SAD_OUT_CTRL_E;
```

【成员】

成员名称	描述
IVE_SAD_OUT_CTRL_16BIT_BOTH	16 bit SAD 图和阈值化图输出模式。
IVE_SAD_OUT_CTRL_8BIT_BOTH	8 bit SAD 图和阈值化图输出模式。
IVE_SAD_OUT_CTRL_16BIT_SAD	16 bit SAD 图输出模式。
IVE_SAD_OUT_CTRL_8BIT_SAD	8 bit SAD 图输出模式。
IVE_SAD_OUT_CTRL_THRESH	阈值化图输出模式。

【注意事项】

无。

【相关数据类型及接口】

IVE\_SAD\_CTRL\_S

4.2.69 IVE\_SAD\_CTRL\_S

【说明】

定义 SAD 控制参数

【定义】

```
typedef struct _IVE_SAD_CTRL_S
{
    IVE_SAD_MODE_E enMode;

    IVE_SAD_OUT_CTRL_E enOutCtrl;

    CVI_U16 u16Thr;

    CVI_U8 u8MinVal;

    CVI_U8 u8MaxVal;
} IVE_SAD_CTRL_S;
```

【成员】



成员名称	描述
enMode	SAD 计算模式。
enOutCtrl	SAD 输出控制模式。
u16Thr	对计算的 SAD 图进行阈值化的阈值。 取值范围依赖 enMode： 1、 IVE_SAD_OUT_CTRL_8BIT_BOTH, 取值 [0, 255] 2、 IVE_SAD_OUT_CTRL_16BIT_BOTH 和 IVE_SAD_OUT_CTRL_THRESH, 取值 [0, 65535]
u8MinVal	阈值化不超过 u16Thr 时的取值。
u8MaxVal	阈值化超过 u16Thr 时的取值

【注意事项】

无。

【相关数据类型及接口】

无。

IVE\_SAD\_MODE\_E

IVE\_SAD\_OUT\_CTRL\_E

4.2.70 IVE\_RESIZE\_MODE\_E

【说明】

定义 Resize 的模式。

【定义】

```
typedef enum _IVE_RESIZE_MODE_E
{
    IVE_RESIZE_MODE_LINEAR = 0x0,
    IVE_RESIZE_MODE_AREA = 0x1,
    IVE_RESIZE_MODE_BUTT
} IVE_RESIZE_MODE_E;
```

**【成员】**

成员名称	描述
IVE_RESIZE_MODE_LINEAR	Resize 线性模式
IVE_RESIZE_MODE_AREA	Resize 区域模式

**【注意事项】**

无。

**【相关数据类型及接口】**

IVE\_RESIZE\_CTRL\_S

## 4.2.71 IVE\_RESIZE\_CTRL\_S

**【说明】**

定义 Resize 控制参数。

**【定义】**

```
typedef struct _IVE_RESIZE_CTRL_S {  
  
    IVE_RESIZE_MODE_E enMode;  
  
    IVE_MEM_INFO_S stMem;  
  
    CVI_U16 u16Num;  
  
} IVE_RESIZE_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	Resize 的模式
stMem	用于 Resize 的内存
u16Num	输入图像数量

**【注意事项】**

无。

**【相关数据类型及接口】**

无。.. \_IVE\_HOG\_CTRL\_S:

## 4.2.72 IVE\_HOG\_CTRL\_S

### 【说明】

定义计算 HOG(Histogram of Oriented Gradient) 特征控制参数。

### 【定义】

```
typedef struct IVE_HOG_CTRL {  
    CVI_U8 u8BinSize;  
  
    CVI_U32 u32CellSize;  
  
    CVI_U16 u16BlkSizeInCell;  
  
    CVI_U16 u16BlkStepX;  
  
    CVI_U16 u16BlkStepY;  
} IVE_HOG_CTRL_S;
```

### 【成员】

成员名称	描述
u8BinSize	每个 Cell 的 histogram bin 个数
u32CellSize	Cell 大小
u16BlkSizeInCell	一个 Cell 包含的 Block size
u16BlkStepX	Stride x
u16BlkStepY	Stride y

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## 4.2.73 IVE\_16BIT\_TO\_8BIT\_MODE\_E

### 【说明】

定义 16BIT 图像数据到 8bit 图像数据的转化模式。

### 【定义】

```
typedef struct _IVE_16BIT_TO_8BIT_MODE_E
```

(下页继续)

(续上页)

```
{
    IVE_16BIT_TO_8BIT_MODE_S16_TO_S8 = 0x0,
    IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS = 0x1,
    IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS = 0x2,
    IVE_16BIT_TO_8BIT_MODE_U16_TO_U8 = 0x3,
    IVE_16BIT_TO_8BIT_MODE_BUTT
}IVE_16BIT_TO_8BIT_MODE_E;
```

【成员】

成员名称	描述
IVE_16BIT_TO_8BIT_MODE_S16_TO_S8	S16 数据到 S8 数据的线性变换。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS	S16 数据线性变换到 S8 数据后取绝对值得到 S8 数据。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS	S16 数据线性变换到 S8 数据且平移后截断到 U8 数据。
IVE_16BIT_TO_8BIT_MODE_S16_TO_U8	S16 数据到 U8 数据的线性变换。

【注意事项】

无。

【相关数据类型及接口】

- IVE\_16BIT\_TO\_8BIT\_CTRL\_S

4.2.74 IVE\_16BIT\_TO\_8BIT\_CTRL\_S

【说明】

定义 16BIT 图像数据到 8bit 图像数据的转化控制参数。

【定义】

```
typedef struct _IVE_16BIT_TO_8BIT_CTRL_S
{
    IVE_16BIT_TO_8BIT_MODE_E enMode;
    CVI_U16 u16Denominator;
```

(下页继续)

(续上页)

```
CVI_U8 u8Numerator;  
  
CVI_S8 s8Bias;  
  
}IVE_16BIT_TO_8BIT_CTRL_S;
```

**【成员】**

成员名称	描述
enMode	16bit 数据到 8bit 数据的转换模式。
u16Denominator	线性变换中的分母。 取值范围：[Max {1, u8Numerator}, 65535]
u8Numerator	线性变换中的分子。 取值范围：[0,255]
s8Bias	线性变换中的平移项。 取值范围：[- 128,127]

**【注意事项】**

无。

**【相关数据类型及接口】**

· IVE\_16BIT\_TO\_8BIT\_MODE\_E

## 4.2.75 IVE\_BLOCK\_CTRL\_S

**【说明】**

IVE\_BLOCK 控制参数。

**【定义】**

```
typedef struct IVE_BLOCK_CTRL {  
  
    CVI_FLOAT f32BinSize;  
  
    CVI_U32 u32CellSize;
```

(下页继续)

(续上页)

```
} IVE_BLOCK_CTRL_S;
```

;

**【成员】**

成员名称	描述
f32Scale	取完 Cell 平均后在除以 Scale value。
u32CellSize	Cell 大小

**【注意事项】**

无。

**【相关数据类型及接口】**

# 5 技巧说明

---

## 5.1 额外的缓冲区

目前仅支持 UINT8/INT8/BF16 的运算，任何超过 UINT8 值域的功能是使用 BF16 实现，速度上会较慢一些，且需要额外的缓冲空间当作暂存区。

# 6 FAQ

---

## 6.1 Cache 内存的使用

内存使用 cache 时机，由算法软件对这内存的使用方式来判断。由于 IVE 是直接读取 DDR 内存数据，此时使用的内存若带有 cache，则必须一直刷 cache 来保证数据的一致性。所以若无频繁处理器操作，那么建议内存不带 cache；反之，建议这片内存带 cache。

## 6.2 bInstant 参数的设定

IVE 各算法函式最后一个参数的介绍，设定 True 为使用 Busy waiting 方式等待中断回应；设定 False 则会将程序移出处理器，等到中断讯号通知，才运行 IVE 中断程序。