CV180X & CV181X IVE API 使用手册

Version: 1.0

Release date: 2022-06-18

目录

修订记录

Version	Date	Description
1.0	2022/06/18	初版

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

晶视智能 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 RISC-V 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。

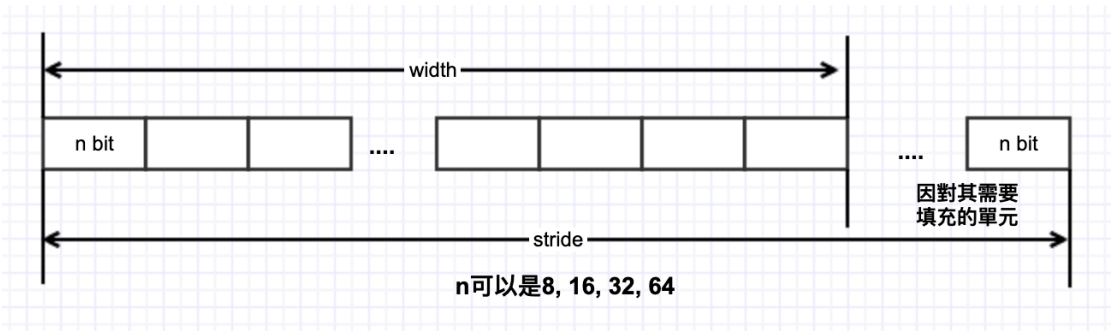
- 跨度 (stride)

  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_U8C1	8bit 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_S8C1	8bit 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_YUV420SP	YUV420 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_YUV422SP	YUV422 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_YUV420P	YUV420 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	YUV422 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_S8C2_PACKAGE	8bit 无符号双通道且以 Package 格式存储的图像	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_S8C2_PLANAR	8bit 无符号双通道且以 Planar 格式存储的图像	内存地址仅用到 IVE_IMAGE_S 中的 u64PhyAddr[0]、u64VirAddr[0],u64PhyAddr[1]、u64VirAddr[1]	跨度仅用到 u32Stride[0]、u32Stride[1]
IVE_IMAGE_TYPE_S16C1	16bit 有符号单通道图像图 1-2	仅用到 IVE_IMAGE_S	仅用到 u32Stride[0]

图 1-2 单通道图像

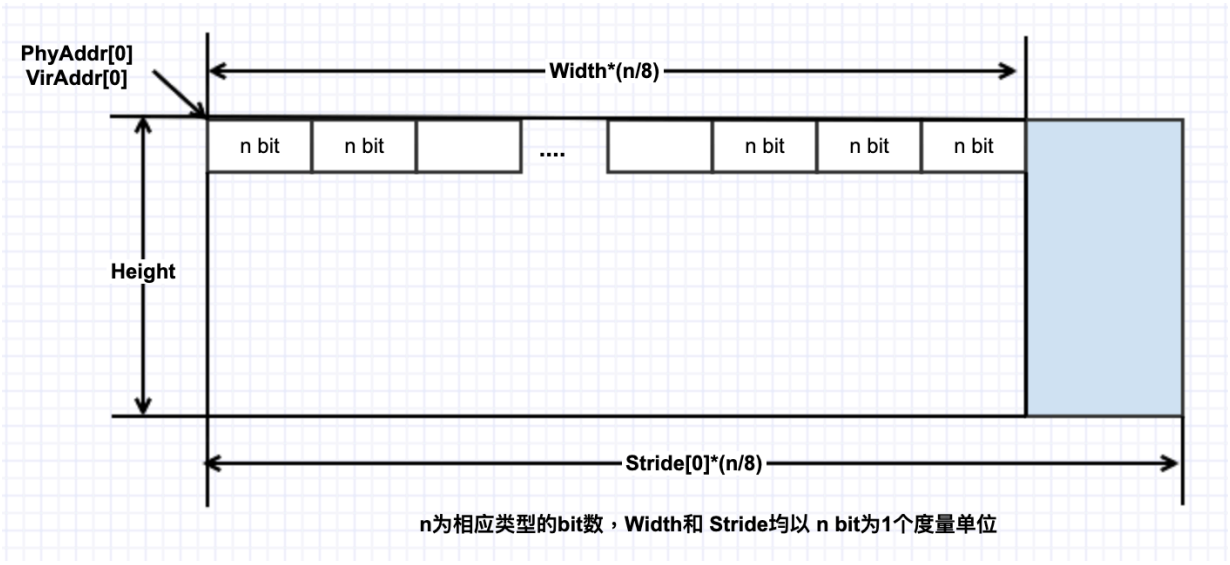


图 1-3 IVE_IMAGE_TYPE_YUV420P 类型的 IVE_IMAGE_S 图像

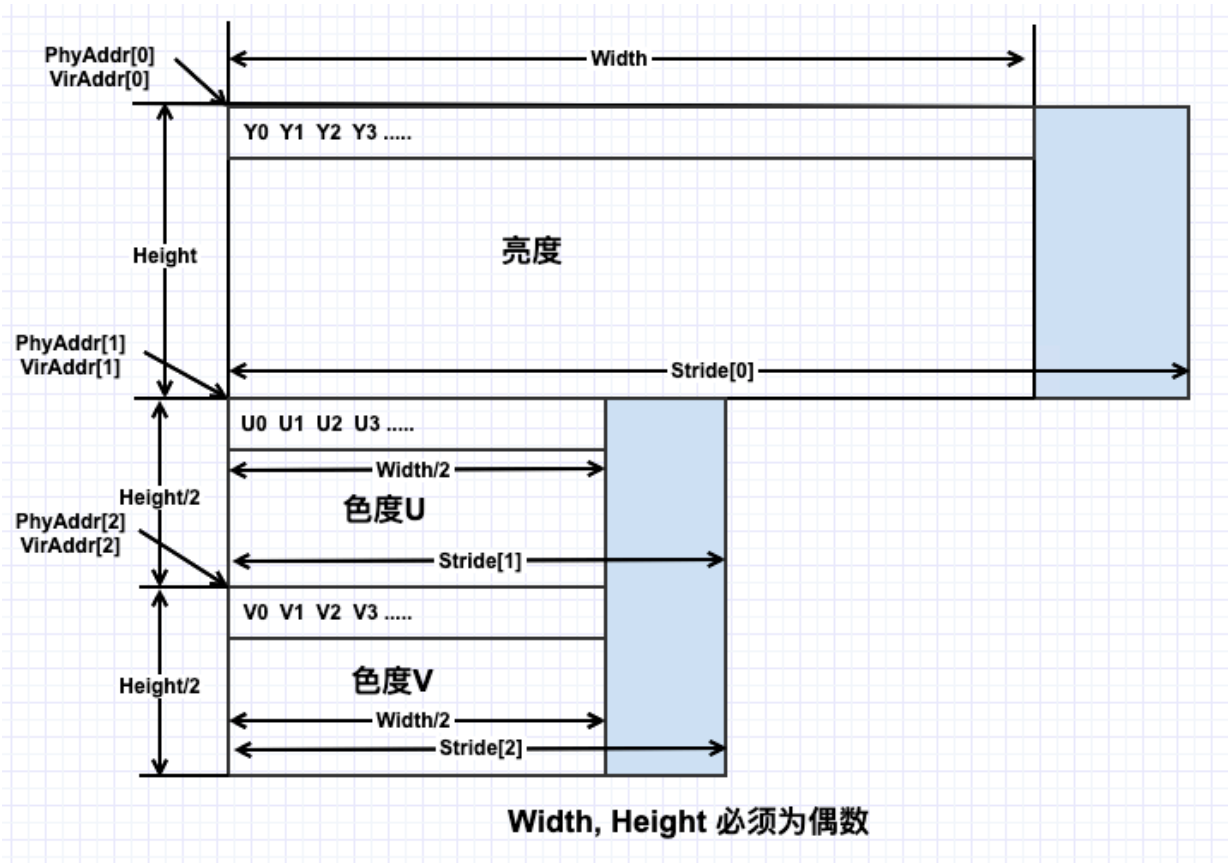


图 1-4 IVE_IMAGE_TYPE_YUV422P 类型的 IVE_IMAGE_S 图像

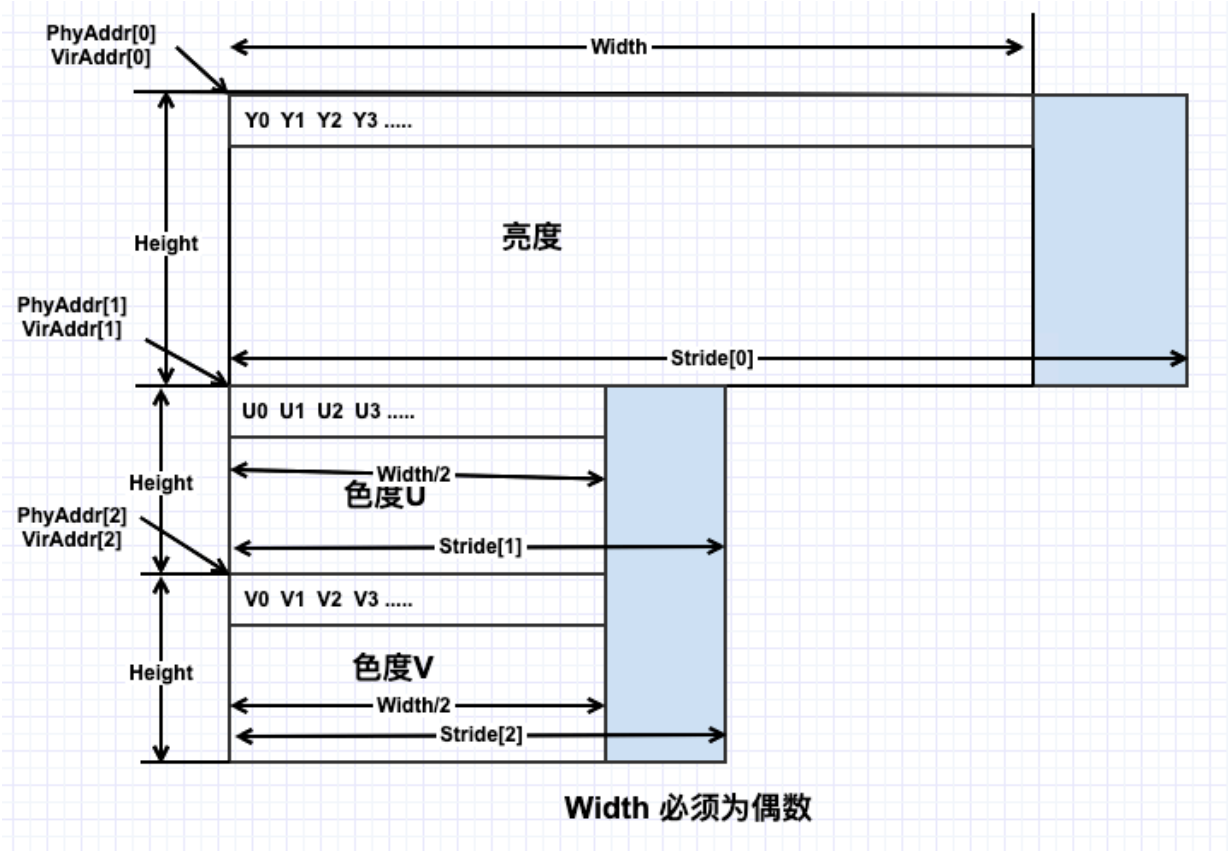


图 1-5 IVE_IMAGE_TYPE_U8C3_PACKAGE 类型的 IVE_IMAGE_S 图像

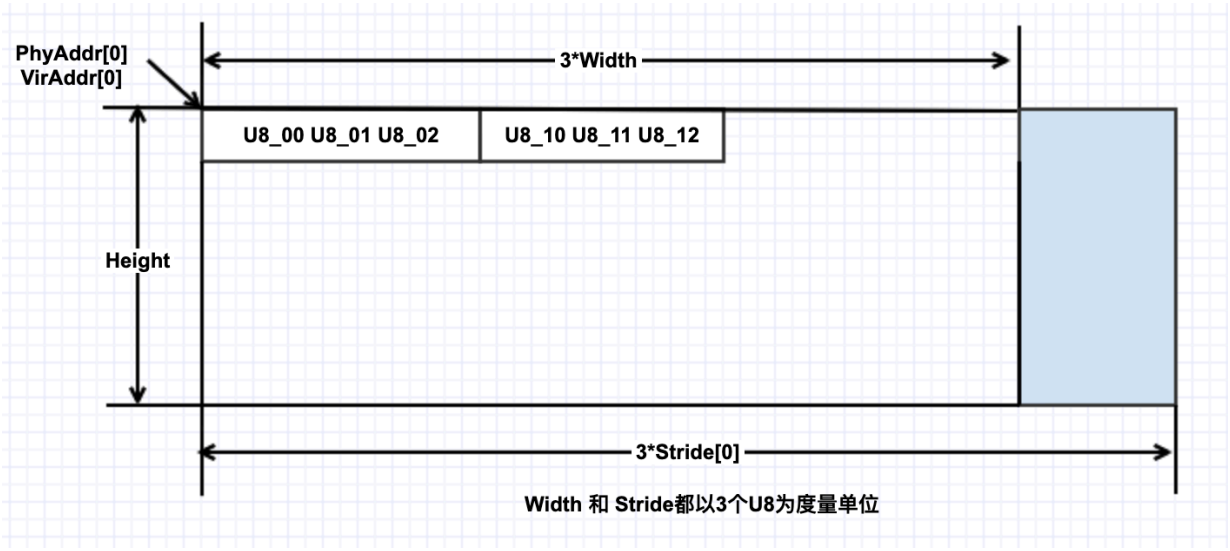
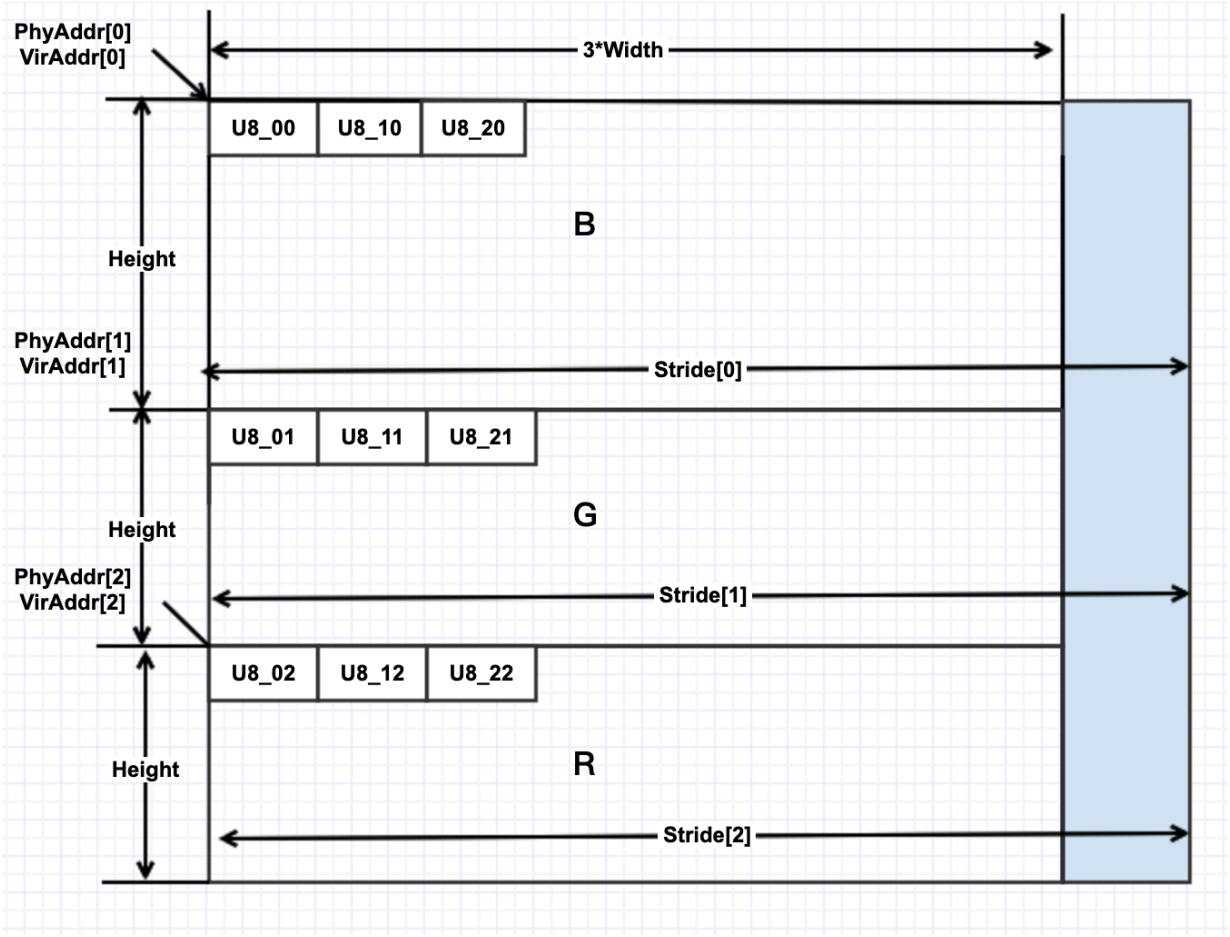


图 1-6 IVE_IMAGE_TYPE_U8C3_PLANAR 类型的 IVE_IMAGE_SRC 图像


PhyAddr[0]
VirAddr[0]
3*Width
U8_00
U8_10
U8_20
B
Height
PhyAddr[1]
VirAddr[1]
Stride[0]
U8_01
U8_11
U8_21
G
Height
PhyAddr[2]
VirAddr[2]
Stride[1]
U8_02
U8_12
U8_22
R
Height
Stride[2]

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_CreateHandle(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_DST_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	源数据指针。不能为空。	输入
pstDst	输出数据指针。Copy 模式下不能为空。	输出
pstCtrl	DMA 控制参数指针。不能为空。	输入
bInstant	返回结果标志。True 为 busy waiting mode,False 为 Interrupt mode。	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 pstSrc; 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

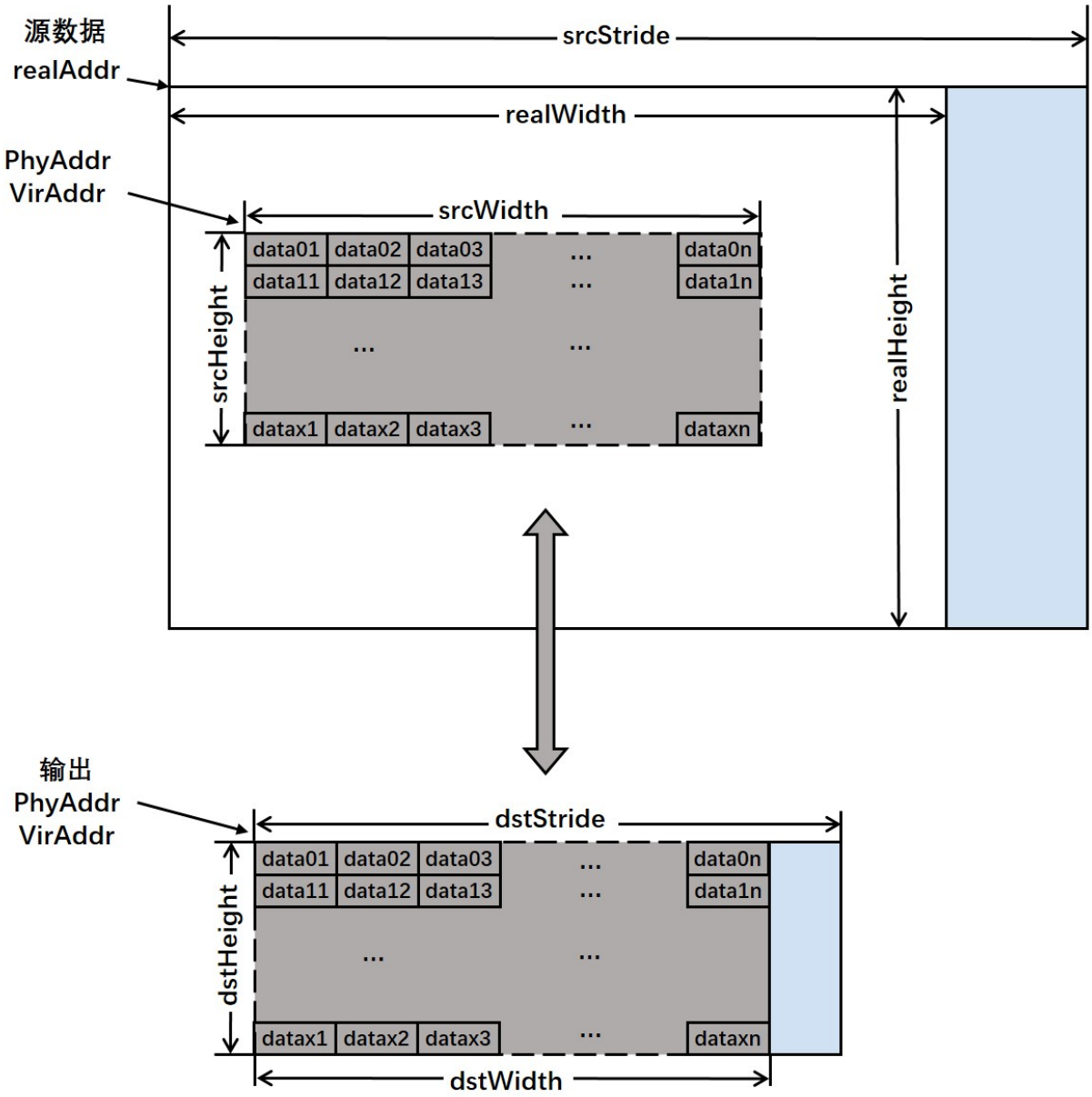
[注意]

· 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：

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

快速拷贝示意图

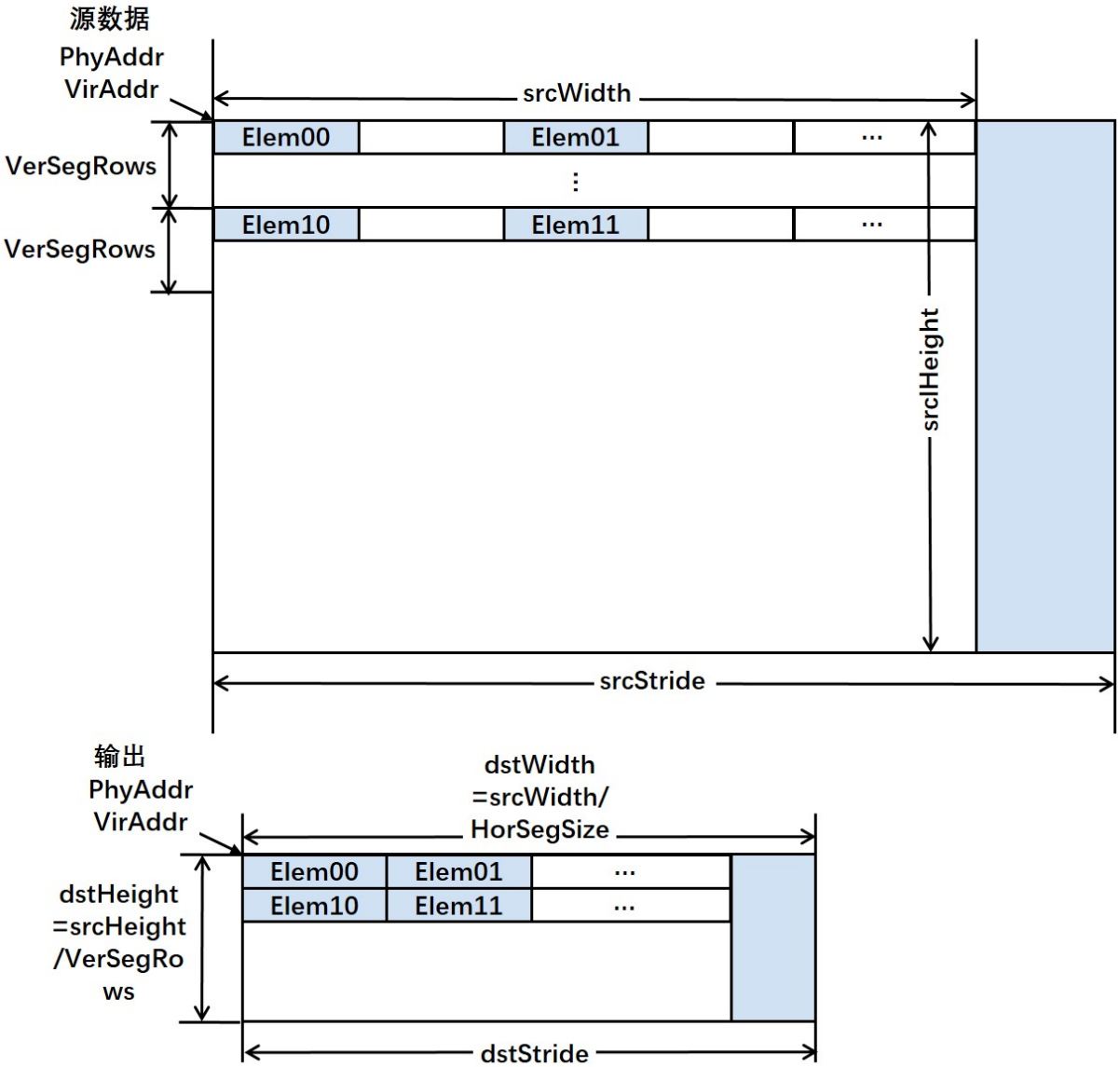


- IVE_DMA_MODE_INTERVAL_COPY: 间隔拷贝模式
  - 要求源数据宽度为 u8HorSegSize 的倍数；
  - 间隔拷贝的方式：将每 u8VerSegRows 行中第一行数据分割为 u8HorSegSize 大小的段，拷贝每段中的前 u8ElemSize 大小的字节。如图所示。
- IVE_DMA_MODE_SET_3BYTE:3 字节填充模式

  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	源数据指针。不能为空。	输入
pstDst	输出数据指针。宽高同 pstSrc。	输出
pstCtrl	控制信息指针。不能为空。	输入
bInstant	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	64x64~1920x1024
pstDst	同 pstSrc	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 模板滤波及 YUV2RGB 色彩空间转换任务，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_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。不能为空。	输入
pstSrc	源数据指针。不能为空。	输入
pstDst	输出数据指针。宽高同 pstSrc。	输出
pstCtrl	控制信息指针。不能为空。	输入
bInstant	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 或 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。

【语法】

```
CVI_S32 CVI_IVE_CSC(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	源数据指针。不能为空。	输入
pstDst	输出数据指针。宽高同 pstSrc。	输出
pstCscCtrl	控制信息指针。不能为空。	输入
bInstant	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	64x64~1920x1024
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	源数据指针。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 pstSobelCtrl→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 pstSobelCtrl→enOutCtrl，若需要输出则不能为空。宽高同 pstSrc。	输出
pstCtrl	控制信息指针。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	8x8~1920x1024
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	源数据指针。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 pstNormGradCtrl→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 pstNormGradCtrl→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。根 pstNormGradCtrl→enOutCtrl，若需要输出则不能为空。宽高同 pstSrc。	输出
pstCtrl	控制信息指针。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	
pstDstH	U8C1	16 byte	
pstDstV	U8C1	16 byte	
pstDstHV	S8C2_PACKAGE	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】

3.9 Canny Edge

【描述】

链接 Canny 影像结边界。

【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 影像，输出二值化边界影像。	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	
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，计算灰阶影像的 Gradient, Gradient Magnitude, Hysteresis threshold 和 Non-Maximum Suppression。

【语法】

```
CVI_S32 CVI_IVE_CannyHysEdge(IVE_HANDLE pIveHandle, IVE_IMAGE_S *pstSrc, IVE_
↪DST_IMAGE_S *pstEdge, IVE_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	源数据指针。不能为空。	输入
pstEdge	Strong/Weak Edge Flag 影像。	输出
pstStack	强边界的坐标	输出
pstCtrl	控制信息指针。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	
pstEdge	U8C1	16 byte	
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	源数据指针。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 pstMagAndAngCtrl→enOutCtrl，需要输出则不能为空。高、宽同 pstSrc。	输出
pstCtrl	控制信息指针。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
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	源数据指针。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	控制信息指针。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	U16C1	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	源数据指针。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	控制信息指针。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 的二值图	16 byte	64x64~1920x1024
pstDst	U8C1 的二值图	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_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	源数据指针。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	控制信息指针。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
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

.

3.15 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, IVE_AND_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	源图像 1 指针。不能为空。	输入
pstSrc2	源图像 2 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
pstCtrl	控制信息指针。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	U16C1 的二值图	1 byte	64x64~1920x1024
pstSrc2	U16C1	1 byte	同 pstSrc
pstDst	U8C1	1 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	源图像 1 指针。不能为空。	输入
pstSrc2	源图像 2 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	控制信息指针。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	U16C1 的二值图	1 byte	64x64~1920x1024
pstSrc2	U16C1	1 byte	同 pstSrc
pstDst	U8C1	1 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, IVE_OR_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	源图像 1 指针。不能为空。	输入
pstSrc2	源图像 2 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	控制信息指针。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	U16C1 的二值图	1 byte	64x64~1920x1024
pstSrc2	U16C1	1 byte	同 pstSrc
pstDst	U8C1	1 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】

将一个影像透过一个映像表格映像到另一个影像。

【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);
```

(下页继续)

(续上页)

【参数】

参数名称	描述	输入/输出
pIveHandle	handle 指针。不能为空。	输入
pstSrc	输入影像指针。不能为空。	输入
pstMap	输入映像表格指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	控制信息指针。不能为空。	输入
bInstant	及时返回结果标志。	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
.

3.19 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	任务的 handle。	输入
pstSrc	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	控制信息指针。不能为空。	输入
bInstant	及时返回结果标志。	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

3.20 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	源数据指针。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
pstIntegCtrl	控制信息指针。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	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	源数据指针。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
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		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	源数据指针。不能为空。	输入
pstSrc2	源数据指针。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	控制信息指针。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		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
.

3.23 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	源图像 1 指针。不能为空。	输入
pstSrc2	源图像 2 指针。不能为空。高宽同 pstSrc1	
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	1 byte	64x64~1920x1024
pstSrc2	U8C1	1 byte	同 pstSrc
pstDst		1 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

3.24 Match BgModel

【描述】

输入当前影像和模型，取得前景数据。

【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	当前影像。	输入
pstBgModel	模型。	输入/输出
pstFgFlag	前景状态影像。	输入/输出
pstDiffFg	前景影像。	输出
pstStatData	前景状态	输出
pstCtrl	控制结构	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	1 byte	
pstBgModel		1 byte	
pstFgFlag	U8C1	1 byte	
pstDiffFg	S8C1	1 byte	
pstStatData		1 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】

更新背景模型。

【语法】

```
CVI_S32 CVI_IVE_UpdateBgModel(IVE_HANDLE pIveHandle, 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_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
pstBgModel	模型	输入/输出
pstFgFlag	前景状态影像。	输入/输出
pstBgImg	背景影像。	输出
pstChaSta	前景更新状态影像。	输出
pstStatData	背景状态	输出
pstCtrl	控制结构	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
pstBgModel		1 byte	
pstFgFlag	U8C1	1 byte	
pstBgImg	U8C1	1 byte	
pstChaSta	S8C1	1 byte	
pstStatData		1 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】

根据背景梯度影像和当前影像，计算前景图梯度影像。

【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	前景影像	输入
pstCurGrad	当前梯度影像	输入
pstBgGrad	背景梯度影像	输入
pstGradFg	前景梯度影像	输出
pstCtrl	控制结构	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	1byte	
pstCurGrad	S8 C2_PACKAGE	1byte	
pstBgGrad	S8 C2_PACKAGE	1byte	
pstGradFg	S8C1	1byte	

【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	输入影像。	输入
pstFg	前景影像。	输出
pstBg	背景影像。	输出
pstModel	模型数据。	输入/输出
pstCtrl	控制结构	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		1byte	
pstSrc	U8C1 或 U8C3_PACKAGE	1byte	
pstFg	U8C1 二值化影像	1 byte	
pstBg	U8 C1 或 U8C3_PACKAGE	1byte	

【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 GMM 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 *pstBg,IVE_DST_
↪IMAGE_S *pstMatchModelInfo, 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	输入影像。	输入
pstFactor	模型更新系数	输入
pstFg	前景影像。	输出
pstBg	背景影像。	输出
pstMatchModelInfo	模型匹配系数	输出
pstModel	模型数据。	输入/输出
pstCtrl	控制结构	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	.	1byte	
pstFactor	U16C1	1byte	
pstSrc	U8C1 或 U8C3_PACKAGE	1byte	
pstFg	U8 C1 二值化影像	1byte	
pstBg	U8C1 或 U8C3_PACKAGE	1byte	
pstMatchModelInfo	U8C1	1byte	

【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	输入影像。	输入
pstDst	结果影像。	输出
pstCtrl	控制结构	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	1byte	
pstDst	U8C1 二值化影像	1 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

3.30 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	源 1 图像指针。不能为空。	输入
pstSrc2	源 2 图像指针。不能为空。	输入
pstDst	输出数据指针。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	1byte	64x64~1920x1024
pstSrc2	U8C1	1byte	同 pstSrc
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

3.31 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	源图像指针。不能为空。	输入
pstDst	输出数据指针。不能为空。内存至少需配置:sizeof(IVE_NCC_DST_MEM_S)。 |	输出
pstCtrl	控制参数指针。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	1byte	64x64~1920x1024
pstSrc2	U8C1	1byte	同 pstSrc
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

3.32 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	源 1 图像指针。不能为空。	输入
pstSrc2	源 2 图像指针。不能为空。高和宽同 pstSrc1	输入
pstSad	输出 SAD 图像指针。 根据 pstSad Ctrl→enOutCtrl，若需要输出则不能为空。 根据 pstSadCtrl→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 pstSad Ctrl→enOutCtrl，若需要输出则不能为空。 根据 pstSadCtrl→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	控制信息指针。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	1byte	64x64~1920x1024
pstSrc2	U8C1	1byte	同 pstSrc1
pstSad	U8C1、U16C1	16byte	根据 pstSadCtrl→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 pstSadCtrl→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

3.33 BufFlush

【描述】

对于使用 CVI_IVE_CreateImage_Cached 建立的图像，在 IVE 硬件存取影像的内容前，须使用此函式将 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
pIveHandle	任务的 handle。	输入
pstImg	操作的影像内容。	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

3.34 BufRequest

【描述】

对于使用 CVI_IVE_CreateImage_Cached 建立的图像，在 RISC-V 存取 u64VirAddr 所指向的内容前，须使用此函式将 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	任务的 handle。	输入
pstImg	操作的影像内容。	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 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	任务的 handle。	输入
pstMemInfo	创建的记忆体结构。不能为空。	输入
u32ByteSize	记忆体结构的 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

3.36 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	任务的 handle。	输入
pstDataInfo	创建的 IVE_DATA_S 结构。不能为空。	输入
u16Width	Data 的宽	输入
u16Height	Data 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

3.37 CreateImage

【描述】

创造一块影像内存供使用。用此函数建立的影像会自动映像 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, CVI_U16 u16Width, CVI_U16 u16Height);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
pstImg	创建给影像用的记忆体结构。	输出
enType	创建的影像记忆体格式	输入
u16Width	影像的宽	输入
u16Height	影像的高	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

3.38 CreateImage with Cache

【描述】

创造一块影像内存供使用。用此函数建立的影像需要使用 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, CVI_U32 u32Width,
CVI_U32 u32Height);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
pstImg	创建给影像用的记忆体结构。	输出
enType	创建的影像记忆体格式	输入
u32Width	影像的宽	输入
u32Height	影像的高	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 ResetImage

【描述】

将 Image 内容填入特定值。

【语法】

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

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
pstImg	创建给影像用的记忆体结构。	输出
val	预填入影像的值	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

3.40 ReadImageArray

【描述】

从 buffer 读入影像。

【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	任务的 handle。	输入
pstImage	创建给影像用的记忆体结构。	输出
pBuffer	Buffer	输入
enType	创建的影像记忆体格式	输入
u16Width	影像的宽	输入
u16Height	影像的高	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

3.41 ReadMem

【描述】

从档案读入到 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	任务的 handle。	输入
pstMem	IVE_MEM_INFO_S 结构。	输出
filename	档案路径	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

3.42 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	任务的 handle。	输入
pstMem	IVE_MEM_INFO_S 结构。	输出
pBuffer	Buffer	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

3.43 ReadData

【描述】

从档案读入到 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	任务的 handle。	输入
pstData	IVE_DATA_S 结构。	输出
filename	档案路径	输入
u16Width	Data 的宽	输入
u16Height	Data 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

3.44 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	任务的 handle。	输入
pstData	IVE_DATA_S 结构。	输出
pBuffer	Buffer	输入
u16Width	Data 的宽	输入
u16Height	Data 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 ReadImage

【描述】

从档案位置读取一张影像。

【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	任务的 handle。	输入
filename	影像档案名称	输入
enType	希望拿到的影像格式	

【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

3.46 ReadRawImage

【描述】

从档案位置读取一张 Raw Image。

【语法】

```
IVE_IMAGE_S CVI_IVE_ReadRawImage(IVE_HANDLE pIveHandle, 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	任务的 handle。	输入
filename	影像档案名称	输入
enType	希望拿到的影像格式	输入
u16Width	影像宽。	输入
u16Height	影像高。	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

3.47 WriteData

【描述】

写入 IVE_DATA_S 内容到档案位置。

【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	任务的 handle。	输入
filename	储存的档案位置及档名	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

3.48 WriteMem

【描述】

写入 IVE_MEM_INFO_S 内容到档案位置。

【语法】

```
CVI_S32 CVI_IVE_WriteData(IVE_HANDLE pIveHandle, const char *filename, IVE_MEM_INFO_
↪S *pstMem);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
filename	储存的档案位置及档名	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

3.49 WriteImage

【描述】

写入一张 PNG 影像到档案位置。

【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	任务的 handle。	输入
filename	储存的档案位置及档名	输入
pstImg	要储存的影像内容	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 WriteRawImage

【描述】

写入一张影像到档案位置。

【语法】

```
CVI_S32 CVI_IVE_WriteImg(IVE_HANDLE pIveHandle, const char *filename, IVE_IMAGE_S␣
↪*pstImg);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
filename	储存的档案位置及档名	输入
pstImg	要储存的影像内容	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

3.51 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	任务的 handle。	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

3.52 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	任务的 handle。	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 Split DiffFg of BgModel

【描述】

从 BgModel 的结果取出 DiffFg，并储存成 YUV 影像。

【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	任务的 handle。	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

3.54 Split ChgSta of BgModel

【描述】

从 BgModel 的结果取出 ChgSta，并储存成 YUV 影像。

【语法】

```
CVI_S32 CVI_IVE_DiffFg_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	任务的 handle。	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

3.55 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	任务的 handle。	输入
pbFinish	回传 task 是否结束	输出
bBlock	True 表示 blocked task	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 Image2VideoFrameInfo

【描述】

将 IVE 影像格式转换成 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	输入的影像内容	输入
pstVFIDst	输出的影像内容	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

3.57 VideoFrameInfo2Image

【描述】

将 Video Frame Info 影像格式转换成 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
pstIISrc	输入的影像内容	输入
pstVFIDst	输出的影像内容	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

3.58 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	任务的 handle。	输入
pstMem	要释放的记忆体结构	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

3.59 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	任务的 handle。	输入
pstImg	输入影像。	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

3.60 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	任务的 handle。	输入
pstData	输入 Data。	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 Thresh_S16

【描述】

创建 S16 数据到 8bit 数据的阀值化任务。

【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	任务的 handle。	输入
pstSrc	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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

3.62 Thresh_U16

【描述】

创建 U16 数据到 8bit 数据的阀值化任务。

【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	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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
-

3.63 Resize

【描述】

创建影像 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	任务的 handle。	输入
astSrc	输入影像数组。不能为空。	输入
astDst	输出的影像数输出组。不能为空。影像类型必须和 astSrc 相同	输出
pstCtrl	门坎值参数结构指针，不能为空。	输入
bInstant	参考值	输出

参数名称	支持图像类型	地址对齐	分辨率
astSrc	U 8C1 或 U8C3_PLANAR	1byte	
astDst	U 8C1 或 U8C3_PLANAR	1byte	

【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

3.64 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, bool bInstant);
```

【参数】

参数名称	描述	输入/输出
pIveHandle	任务的 handle。	输入
pstSrc	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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

3.65 RGB YUV Erode to Dilate

【描述】

【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	任务的 handle。	输入
pstSrc	输入影像指针。不能为空。	输入
pstDst1	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstDst2	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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

3.66 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	任务的 handle。	输入
pstSrc	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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

3.67 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	任务的 handle。	输入
pstSrc1	输入影像指针。不能为空。	输入
pstSrc2	输入影像指针。不能为空。	输入
pstDst	输出的影像指针。不能为空。高和宽同 pstSrc。	输出
pstCtrl	门坎值参数结构指针，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

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_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_MAG_DIST_E: 定义梯度幅值距离计算方式。
- 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 16bti 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_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_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_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_HOG_CTRL_S: 定义计算 HOG(Histogram of Oriented Gradient) 特征控制参数。
- IVE_GRAD_FG_CTRL_S: 定义 Gradfg 控制参数。
- IVE_GRAD_FG_MODE_E: 定义 Gradfg 的模式。

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_NUM

【说明】

定义二维广义图像支持的图像类型。

【定义】

```
typedef enum IVE_IMAGE_TYPE {
  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,
```

(下页继续)

(续上页)

```
  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

{

  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
{
  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	无号 16bit X 坐标。
u16Y	无号 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;
```

(下页继续)

(续上页)

```
} IVE_POINT_S16_S;
```

【成员】

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

【注意事项】

无。

【相关数据类型及接口】

无。

4.2.14 IVE_DMA_MODE_E

【说明】

定义 DMA 操作模式。

【定义】

```
typedef struct IVE_DMA_MODE

{

  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。

4.2.15 IVE_DMA_CTRL_S

【说明】

定义 DMA 控制信息。

【定义】

```
typedef struct IVE_DMA_CTRL
{
  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

4.2.16 IVE_FILTER_CTRL_S

【说明】

定义模板滤波控制信息。

【定义】

```
typedef struct IVE_FILTER_CTRL

{

  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。

4.2.17 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,
```

(下页继续)

(续上页)

```
  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_YUV2RGB	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

4.2.18 IVE_CSC_CTRL_S

【说明】

定义色彩空间转换控制信息。

【定义】

```
typedef struct cviIVE_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

4.2.19 IVE_SOBEL_OUT_CTRL_E

【说明】

定义 Sobel 输出控制信息。

【定义】

```
typedef enum IVE_SOBEL_OUT_CTRL

{

  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

4.2.20 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
{
  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。
U8MaskSize	Mask Size
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

4.2.21 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
{
  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

4.2.22 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
{
  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	阀值
as8Mask	5x5 Filter

【注意事项】

无。

【相关数据类型及接口】

IVE_MAG_AND_ANG_OUT_CTRL_E

4.2.23 IVE_DILATE_CTRL_S

【说明】

定义控制信息。

【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。

4.2.24 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。

4.2.25 IVE_THRESH_MODE_E

【说明】

定义图像二值化输出格式。

【定义】

```
typedef enum IVE_THRESH_MODE
{
    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	srcVal $\leqslant$ lowThr, dstVal = minVal; srcVal > lowThr, dstVal = maxVal。
IVE_THRESH_MODE_TRUNC	srcVal $\leqslant$ lowThr, dstVal = srcVal srcVal > lowThr, dstVal = maxVal
IVE_THRESH_MODE_TO_MINVAL	srcVal $\leqslant$ lowThr, dstVal = minVal srcVal > lowThr, dstVal = srcVal
IVE_THRESH_MODE_MIN_MID_MAX	srcVal $\leqslant$ lowThr, dstVal = minVal lowThr < srcVal $\leqslant$ highThr, dstVal = midVal srcVal > highThr, dstVal = maxVal
IVE_THRESH_MODE_ORI_MID_MAX	srcVal $\leqslant$ lowThr, dstVal = srcVal lowThr < srcVal $\leqslant$ highThr dstVal = midVal srcVal > highThr, dstVal = maxVal
IVE_THRESH_MODE_MIN_MID_ORI	srcVal $\leqslant$ lowThr, dstVal = minVal lowThr < srcVal $\leqslant$ highThr dstVal = midVal srcVal > highThr, dstVal = srcVal
IVE_THRESH_MODE_MIN_ORI_MAX	srcVal $\leqslant$ lowThr, dstVal = minVal lowThr < srcVal $\leqslant$ highThr dstVal = srcVal srcVal > highThr, dstVal = maxVal
IVE_THRESH_MODE_ORI_MID_ORI	srcVal $\leqslant$ lowThr, dstVal = srcVal lowThr < srcVal $\leqslant$ highThr dstVal = midVal srcVal > highThr, dstVal = srcVal

【注意事项】

无。

【相关数据类型及接口】

IVE_THRESH_CTRL_S

4.2.26 IVE_THRESH_CTRL_S

【说明】

定义图像二值化控制信息。

【定义】

```
typedef struct IVE_THRESH_CTRL
{
  CVI_U32 enMode;
  CVI_U8 u8LowThr;
  CVI_U8 u8HighThr;
  CVI_U8 u8MinVal;
```

(下页继续)

(续上页)

```
  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

4.2.27 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

4.2.28 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

4.2.29 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,
```

(下页继续)

(续上页)

```
  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

4.2.30 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

4.2.31 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;
} IVE_THRESH_S16_MODE_E;
```

【成员】

成员名称	描述
IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
IVE_THRESH_S16_MODE_S16_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_S16_CTRL_S

4.2.32 IVE_THRESH_S16_CTRL_S

【说明】

定义 16bit 有符号图像的阀值化控制参数。

【定义】

```
typedef struct IVE_THRESH_S16_ CTRL
{
  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

4.2.33 IVE_THRESH_U16_MODE_E

【说明】

定义 16bit 无符号图像的阀值化控制参数。

【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

4.2.34 IVE_THRESH_U16_CTRL_S

【说明】

定义 16bit 无符号图像的阀值化控制参数。

【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;
```

(下页继续)

(续上页)

```
} cviIVE_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

4.2.35 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 cviIVE_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

4.2.36 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 cviIVE_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

4.2.37 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
{
  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

4.2.38 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 cviIVE_ORD_STAT_FILTER_CTRL_S
{
  IVE_ORD_STAT_FILTER _MODE_E enMode;
```

(下页继续)

(续上页)

```
} 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

4.2.39 IVE_MAP_MODE_E

【说明】

MAP 模式。

【定义】

```
typedef enum _IVE_MAP_CODE_E
{
  IVE_MAP_MODE_U8 = 0x0;
  IVE_MAP_MODE_S16 = 0x1;
  IVE_MAP_MODE_U16 = 0x2;
} IVE_MAP_CODE_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】

无。

4.2.40 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”
u0q16X	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。

4.2.41 IVE_NCC_DST_MEM_S

【说明】

定义 NCC 的输出内存信息。

【定义】

```
typedef struct cviIVE_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。

4.2.42 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;
  CVI_U0Q16 u0q16InitWeight;
```

(下页继续)

(续上页)

```
  CVI_U8 u8ModelNum;

} IVE_GMM_CTRL_S;
```

【成员】

成员名称	描述
u22q10NoiseVar	初始噪声变异数数值范围：[0x1, 0xFFFFFF]
u22q10MaxVar	模型最大变异数数值范围：[0x1, 0xFFFFFF]
u22q10MinVar	模型最小变异数数值范围：[1, u22q10MaxVar]
u0q16LearnRate	学习率数值范围：[1, 65535]
u0q16BgRatio	背景比率阀值数值范围：[1, 65535]
u8q8VarThr	变异数阀值数值范围：[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。

4.2.43 IVE_LBP_CMP_MODE_E

【说明】

定义 LBP 计算的比较模式。

【定义】

```
typedef enum cviIVE_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。

4.2.44 IVE_LBP_CTRL_S

【说明】

定义 LBP 纹理计算控制参数。

【定义】

```
typedef struct cviIVE_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

4.2.45 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 cviIVE_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

4.2.46 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 {
  IVE_NORM_GRAD_OUT_CTRL_E enOutCtrl;
  IVE_MAG_DIST_E enDistCtrl;
```

(下页继续)

(续上页)

```
  IVE_ITC_TYPE_E enITCType;

  CVI_U8 u8MaskSize;

} IVE_NORM_GRAD_CTRL_S;
```

【成员】

成员名称	描述
enOutCtrl	输出格式
enDistCtrl	计算距离方式
enITCType	是否做归一化
u8MaskSize	屏蔽大小

【注意事项】

无。

【相关数据类型及接口】

IVE_ITC_CTRL_S

IVE_NORM_GRAD_OUT_CTRL_E

4.2.47 IVE_SAD_MODE_E

【说明】

定义 SAD 计算模式。

【定义】

```
typedef enum cviIVE_SAD_MODE_E

{

  IVE_SAD_MODE_MB_4x4 = 0x0,

  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_4x4	按 8x8 像素块计算 SAD。
IVE_SAD_MODE_MB_4x4	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

4.2.48 IVE_SAD_OUT_CTRL_E

【说明】

定义 SAD 输出模式。

【定义】

```
typedef enum cviIVE_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

4.2.49 IVE_SAD_CTRL_S

【说明】

定义 SAD 控制参数

【定义】

```
typedef struct cviIVE_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】

IVE_SAD_MODE_E

IVE_SAD_OUT_CTRL_E

4.2.50 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。

4.2.51 IVE_GRAD_FG_CTRL_S

【说明】

定义 Gradfg 控制参数。

【定义】

```
typedef struct _IVE_GRAD_FG_CTRL_S {
    IVE_GRAD_FG_MODE_E enMode;
    CVI_U16 u16EdwFactor;
```

(下页继续)

(续上页)

```
    CVI_U8 u8CrlCoefThr;

    CVI_U8 u8MagCrlThr;

    CVI_U8 u8MinMagDiff;

    CVI_U8 u8NoiseVal;

    CVI_U8 u8EdwDark;

} IVE_GRAD_FG_CTRL_S;
```

【成员】

成员名称	描述
enMode	Gradfg 的模式
u16EdwFactor	边缘宽度调整因子（范围：500 至 2000；默认值：1000）
u8CrlCoefThr	梯度向量相关系数阈值（范围：50 至 100；默认值：80）
u8MagCrlThr	梯度幅度阈值（范围：0 至 20；默认值：4）
u8MinMagDiff	梯度幅度差异阈值（范围：2 至 8；默认值：2）
u8NoiseVal	梯度幅度噪声阈值（范围：1 至 8；默认值：1）
u8EdwDark	黑色像素启用标志（范围：0（否），1（是）；默认值：1）

【注意事项】

无。

【相关数据类型及接口】

IVE_GRAD_FG_MODE_E

4.2.52 IVE_GRAD_FG_MODE_E

【说明】

定义 Gradfg 的模式。

【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_MODE_USE_CUR_GRAD	当前位置梯度计算模式。
IVE_GRAD_FG_MODE_FIND_MIN_GRAD	周边最小梯度计算模式。

【注意事项】

无。

【相关数据类型及接口】

无。

4.2.53 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 cviIVE_16BIT_TO_8BIT_CTRL_S
{
  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

4.2.54 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 cviIVE_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

4.2.55 IVE_IVE_TYPE_E

【说明】

归一化参数。

【定义】

```
typedef enum IVE_ITC_TYPE {
  IVE_ITC_SATURATE = 0x0,
  IVE_ITC_NORMALIZE = 0x1,
} IVE_ITC_TYPE_E;
```

【成员】

成员名称	描述
IVE_ITC_SATURATE	饱和。
IVE_ITC_NORMALIZE	归一化。

【注意事项】

无。

【相关数据类型及接口】

- IVE_ITC_CTRL_S
- IVE_NORM_GRAD_CTRL_S

4.2.56 IVE_IVE_CTRL_S

【说明】

图像格式转换参数。

【定义】

```
typedef struct IVE_ITC_CRTL {
  IVE_ITC_TYPE_E enType;
} IVE_ITC_CRTL_S;
```

【成员】

成员名称	描述
enType	归一化参数。

【注意事项】

无。

【相关数据类型及接口】

· IVE_ITC_TYPE_E

4.2.57 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;
} 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】

· IVE_ITC_TYPE_E

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。所以如果是无频繁 RISC-V 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 RISC-V, 等到中断讯号通知, 才运行 IVE 中断程序。