MI IVE API

Version 2.05

REVISION HISTORY

Revision No.	Description	Date
2.03	• Initial release	04/12/2018
2.04	• Added function descriptions in regard to MI_IVE_Bernsen, MI_IVE_LineFilterHor, MI_IVE_LineFilterVer, MI_IVE_NoiseRemoveHor, MI_IVE_NoiseRemoveVer, MI_IVE_AdpThresh, MI_IVE_Resize, MI_IVE_BAT, and MI_IVE_Acc	06/05/2019
2.05	• Added description of functions from list matrix_transform and image_dot • Added description of new input parameter for add function • Added description of new input mode for lbp function	10/09/2019

TABLE OF CONTENTS

1. API 参考

该功能提供以下 API：

API名	功能
	建立IVE handle
a	释放IVE handle
MI_IVE_Filter	执行 5x5 模板滤波任务
MI_IVE_Csc	执行色彩空间转换任务
MI_IVE_FilterAndCsc	执行模板滤波加色彩空间转换复合任务
MI_IVE_Sobel	执行 5x5 模板 Sobel-like 梯度计算任务。
MI_IVE_MagAndAng	执行 5x5 模板计算梯度幅值与幅角任务。
MI_IVE_Dilate	执行膨胀任务
MI_IVE_Erode	执行腐蚀任务。
MI_IVE_Thresh	执行图像二值化任务。
MI_IVE_And	执行两图像相与任务
MI_IVE_Sub	执行两图像相减任务。
MI_IVE_Or	执行两图像相或任务
MI_IVE_Integ	执行积分图统计任务。
MI_IVE_Hist	执行直方图统计任务。
MI_IVE_ThreshS16	执行 S16 数据到 8bit 数据阈值化任务
MI_IVE_ThreshU16	执行 U16 数据到 U8 数据阈值化任务。
MI_IVE_16BitTo8Bit	执行 16bit 数据到 8bit 数据线性转化任务。
MI_IVE_OrdStatFilter	执行 3x3 模板顺序统计量滤波任务
MI_IVE_Map	执行 Map（映射 U8->U8 赋值）任务。
MI_IVE_EqualizeHist	执行灰度图像的直方图均衡化计算任务
MI_IVE_Add	执行两灰度图像的加权加计算任务
MI_IVE_Xor	执行两二值图的异或计算任务
MI_IVE_Ncc	执行两相同分辨率图像的归一化互相关系数计算任务
MI_IVE_Ccl	执行二值图像的连通区域标记任务

API名	功能
MI_IVE_Gmm	执行 GMM 背景建模任务
MI_IVE_CannyHysEdge	执行灰度图的 Canny 强弱边缘提取任务
MI_IVE_CannyEdge	灰度图的 Canny 边缘提取的后半部：连接边缘点，形成Canny 边缘图。
MI_IVE_Lbp	执 行LBP 计算任务。
MI_IVE_NormGrad	执行归一化梯度计算任务，梯度均分量均归一化到 S8
MI_IVE_LkOpticalFlow	执行单层 LK 光流计算任务。
MI_IVE_Sad	计算两幅图像按 4x4\8x8\16x16 分块的 16 bit\8 bit SAD 图像， 以及对 SAD 进行阈值化输出。
MI_IVE_Bernsen	执行 3x3 和 5x5 模板之Bernsen门坎值任务。
MI_IVE_LineFilterHor	针对二位图像进行水平方向的滤波任务
MI_IVE_LineFilterVer	针对二位图像进行垂直方向的滤波任务
MI_IVE_NoiseRemoveHor	针对二位图像进行水平方向之噪声滤除任务。
MI_IVE_NoiseRemoveVer	针对二位图像进行垂直方向之噪声滤除任务。
MI_IVE_AdpThresh	执行使用自适应性门坎值的二值化任务。
MI_IVE_Resize	执行缩放影像任务。
MI_IVE_BAT	针对二位图像执行水平和垂直方向的数值交替次数计算任务。
MI_IVE_Acc	执行两灰度图像的累积运算任务。
MI_IVE_Matrix_Transform	执行矩阵运算任务。
MI_IVE_Image_Dot	执行图像点乘积任务。

1.1. MI_IVE_Create

- 功能

  建立IVE handle

- 语法

  MI_IVE_HANDLE MI_IVE_Create(MI_IVE_HANDLE hHandle);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 必须是未使用的hHandle 号 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

- 依赖

  - 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
  - 库文件：libive.a

1.2. MI_IVE_Destroy

- 功能

  释放IVE handle

- 语法

  MI_IVE_HANDLE MI_IVE_Destroy(MI_IVE_HANDLE hHandle);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

1.3. MI_IVE_Filter

➢ 功能

执行 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

MI_S32 MI_IVE_Filter(
MI_IVE_HANDLE hHandle,
MI_IVE_SrcImage_t *pstSrc,
MI_IVE_DstImage_t *pstDst,
MI_IVE_FilterCtrl_t *pstFltCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstFltCtrl	控制信息指针。不能为空。	输入
bInstant	保留	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

注：U8C1\YUV420SP\YUV422SP 均为MI_IVE_ImageType_e 成员的简写，后续其他的成员在表述中也用相同的规则简写。

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

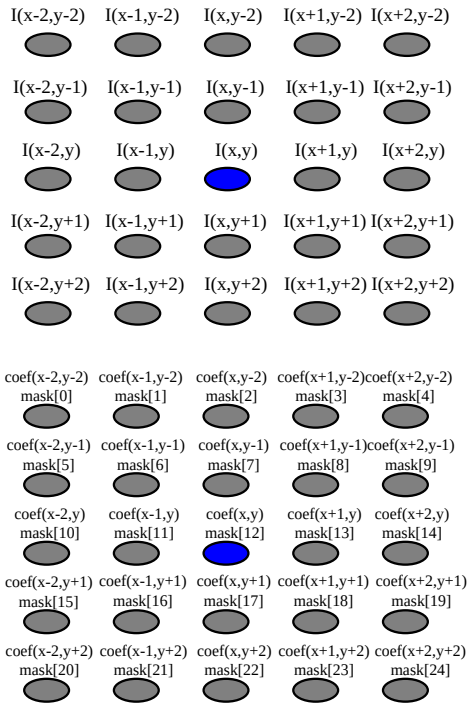
- 依赖

  - 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
  - 库文件：libive.a

- 注意

  - 当源数据为 YUV420SP、YUV422SP 类型时，要求输出数据跨度一致。
  - Filter 计算公式示意如图2-3 所示。

图2-3 Filter 计算公式示意图



$$I_{out}(x,y)=\{\sum_{-2\le i\le 2}\sum_{-2\le j\le 2} I_{in}(x+i,y+j)*coef(x+i,y+j)\}>>norm$$

其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应 pstDst，$coef$（mask）对应 pstFltCtrl 中的as8Mask[MI_IVE_MASK_SIZE_5X5]，norm对应pstFltCtrl中的u8Norm。

- 经典高斯模板如下。

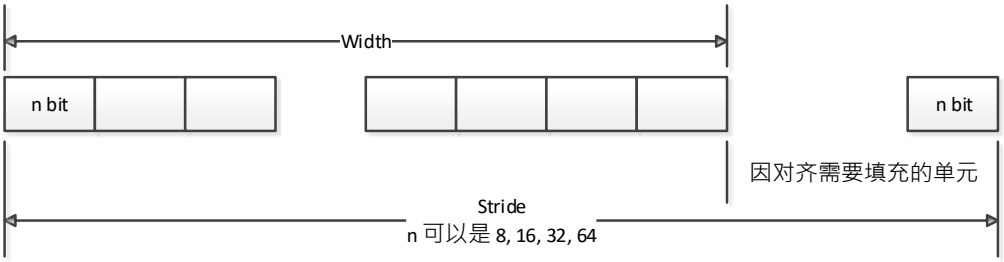
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 2 & 4 & 2 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 5 & 6 & 5 & 2 \\ 3 & 6 & 8 & 6 & 3 \\ 2 & 5 & 6 & 5 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} * 3 \quad \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

$u8Norm = 4$ $u8Norm = 8$ $u8Norm = 8$

MI_IVE_Data_t 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。

可以将 MI_IVE_Data_t 看成一个“像素”用 8bit 表示的图像，那么跨度即统一表述为图像或二维数据的一行以“像素”计算的单元个数。

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 算子对其输入输出有要求 1byte 对齐、2byte 对齐以及 16byte 对齐的，具体见各算子 API 参考中的参数要求。

  - 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（具体结构定义请参见“3 数据类型”）

  - 二维广义图像数据

    MI_IVE_Image_t、MI_IVE_SrcImage_t、MI_IVE_DstImage_t，图像的类型参考 MI_IVE_ImageType_e，具体的内存分配如图 1-2～图 1-10 所示。

    注意：当前所有算子输入输出的二维广义图像数据的高宽均需为偶数。

  - 二维单分量数据

    MI_IVE_Data_t，以 byte 为单位的二维数据，主要用于 DMA 等，其内存如图 1-11 所示；根据类型 MI_IVE_Image_t 可以转化为单个或多个 MI_IVE_Data_t。

– 一维数据

MI_IVE_MemInfo_t、MI_IVE_SrcMemInfo_t、MI_IVE_DstMemInfo_t，表示一维数据，如 Hist 的统计数据、GMM 的模型数据、LKOpticalFlow 的角点输入等；其内存如图 1-12 所示。

- 二维广义图像类型

类型	图像描述	内存地址	跨度
E_MI_IVE_Image_tYPE_U8C1	8bit 无符号单通道图像，如图1-2 所示	仅用到 MI_IVE_Image_t 中的 u32PhyAddr[0]、pu8VirAddr[0]	仅用到 u16Stride[0]

➢ 举例

无。

➢ 相关主题

- MI_IVE_FilterAndCsc
- MI_IVE_OrdStatFilter

1.4. MI_IVE_Csc

➢ 功能

执行色彩空间转换任务，可实现 YUV2RGB\YUV2HSV\YUV2LAB\RGB2YUV 的色彩空间转换。

➢ 语法

MI_S32 MI_IVE_Csc(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_CscCtrl_t *pstCscCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstCscCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

- 当输出数据为 U8C3_PLANAR、YUV420SP、YUV422SP 类型时，要求输出数据跨度一致。
- 支持 12 种工作模式，不同的模式其输出的取值范围不一样，具体请参见 MI_IVE_CscMode_e。
- YUV2HSV、YUV2LAB 参考 OpenCV 中的实现方法。
- 本文档中所提到的 OpenCV，均指 OpenCV 2.4.8 版本。

➢ 举例

无。

➢ 相关主题

MI_IVE_FilterAndCsc

1.5. MI_IVE_FilterAndCsc

➢ 功能

执行 5x5 模板滤波和 YUV2RGB 色彩空间转换复合任务，通过一次执行完成两种功能。

➢ 语法

MI_S32 MI_IVE_FilterAndCsc(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_FilterAndCscCtrl_t *pstFltCscCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输图像据指针。不能为空。 高、宽同 pstSrc。	输出
pstFltCscCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

当输出数据为 U8C3_PLANAR 类型时，要求输出数据跨度一致。

仅支持 YUV2RGB 的4 种工作模式，具体参见 MI_IVE_CscMode_e。

➢ 举例

无。

➢ 相关主题

MI_IVE_Filter

1.6. MI_IVE_Sobel

- 功能

执行 5x5 模板 Sobel-like 梯度计算任务。

- 语法

MI_S32 MI_IVE_Sobel(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDstH, MI_IVE_DstImage_t *pstDstV, MI_IVE_SobelCtrl_t *pstSobelCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。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 pstSobelCtrl→eOutCtrl，若需要输出则不能为空。高、宽同 pstSrc。	输出
pstDstV	由转置后的模板滤波得到的梯度分量图像 V 指针。根据 pstSobelCtrl→eOutCtrl，若需要输出则不能为空。 高、宽同 pstSrc。	输出
pstSobelCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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
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，参见错误码。

- 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

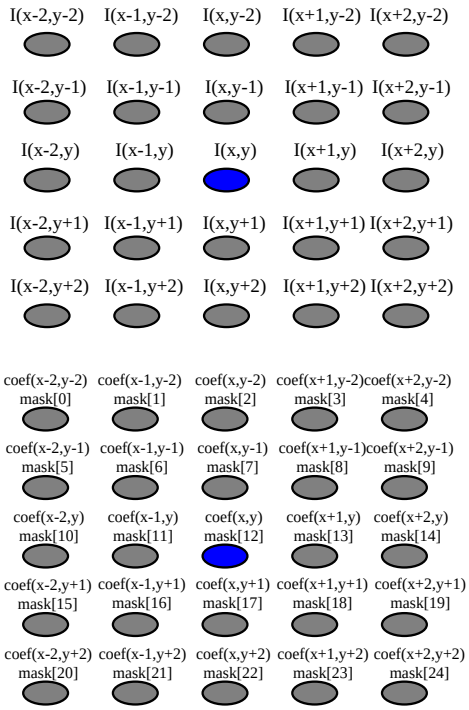
※ 注意

可配置 3 种输出模式，参考 MI_IVE_SobelOutCtrl_e。

当输出模式为 E_MI_IVE_SOBEL_OUT_CTRL_BOTH 时，要求 pstDstH 和pstDstV 跨度一致。

Sobel 计算公式示意如图2-4 所示。

图2-4 Sobel 计算公式示意图



$$H_{out}(x,y)=\sum_{-2\le i\le 2}\sum_{-2\le j\le 2}I(x+i,y+j)*coef(x+i,y+j)$$

$$V_{out}(x,y)=\sum_{-2\le i\le 2}\sum_{-2\le j\le 2}I(x+i,y+j)*coef(x+i,y+j)$$

其中，$I(x, y)$ 对应 pstSrc，$Hout(x, y)$ 对应 pstDstH，$Vout(x, y)$ 对应 pstDstV，$coef$ (mask) 为 pstSobelCtrl 中的as8Mask[MI_IVE_MASK_SIZE_5X5]

- Sobel 模板

$$\begin{bmatrix}0&0&0&0&0\\0&-1&0&1&0\\0&-2&0&2&0\\0&-1&0&1&0\\0&0&0&0&0\end{bmatrix}\quad\begin{bmatrix}0&0&0&0&0\\0&-1&-2&-1&0\\0&0&0&0&0\\0&1&2&1&0\\0&0&0&0&0\end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

- Scharr 模板

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & 0 & 3 & 0 \\ 0 & -10 & 0 & 10 & 0 \\ 0 & -3 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -3 & -10 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 10 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- 拉普拉斯模板

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -4 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & -8 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & -1 & 8 & -1 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

➢ 举例

无。

➢ 相关主题

MI_IVE_MagAndAng

MI_IVE_NormGrad

1.7. MI_IVE_MagAndAng

➢ 功能

执行 5x5 模板梯度幅值与幅角计算任务。

➢ 语法

MI_S32 MI_IVE_MagAndAng(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDstMag, MI_IVE_DstImage_t *pstDstAng, MI_IVE_MagAndAngCtrl_t*pstMagAndAngCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。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 pstMagAndAngCtrl→eOutCtrl，需要输出则不能为空。 高、宽同 pstSrc。	输出
pstMagAndAngCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

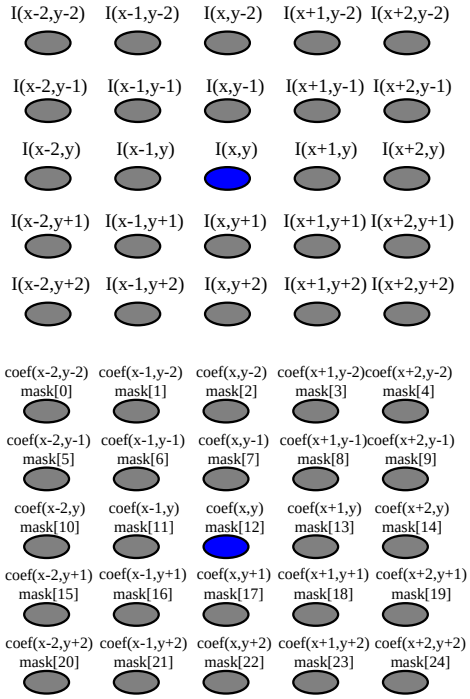
※ 注意

- 可配置 2 种输出模式，具体参见 MI_IVE_MagAndAngOutCtrl_e。
- 当输出模式为 E_MI_IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG 时，要求 pstDstMag和 pstDstAng跨度一致。
- 用户可以通过pstMagAndAngCtrl→u16Thr对幅值图进行thresh 操作(可以用来实现 EOH)，计算公式如下：

$$Mag(x,y)=\begin{cases}0 & Mag(x,y)<\text{u16Thr}\\ Mag(x,y) & Mag(x,y)\geq\text{u16Thr}\end{cases}$$

其中， *Mag*(*x*, *y*) 对应 pstDstMag。

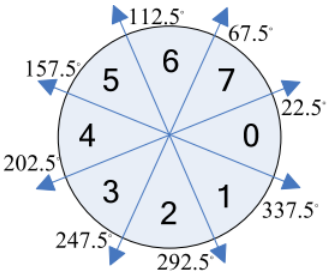
图2-5 MagAndAng 计算示意图



$$H_{out}(x,y)=\sum_{-2\le i\le 2}\sum_{-2\le j\le 2} I(x+i,y+j)*coef(x+i,y+j)$$

$$V_{out}(x,y)=\sum_{-2\le i\le 2}\sum_{-2\le j\le 2} I(x+i,y+j)*coef(x+i,y+j)$$

$$Mag(x,y)=abs(H_{out}(x,y)+abs(V_{out}(x,y))$$



θ(x,y)根据Hout(x,y)、Vout(x,y)以及arctan(Vout/Hout)取对应上图中0～7的方向值。

其中，I(x,y)对应pstSrc，Mag(x,y)对应pstDstMag，θ(x,y)对应pstDstAng，coef(mask)为pstMagAndAngCtrl中的as8Mask[MI_IVE_MASK_SIZE_5X5]。

- 举例

无。

➢ 相关主题

- MI_IVE_CannyHysEdge
- MI_IVE_CannyEdge
- MI_IVE_Sobel

1.8. MI_IVE_Dilate

➢ 功能

执行二值图像 5x5 模板膨胀任务。

➢ 语法

MI_S32 MI_IVE_Dilate(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_DilateCtrl_t *pstDilateCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstDilateCtrl	控制信息指针。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

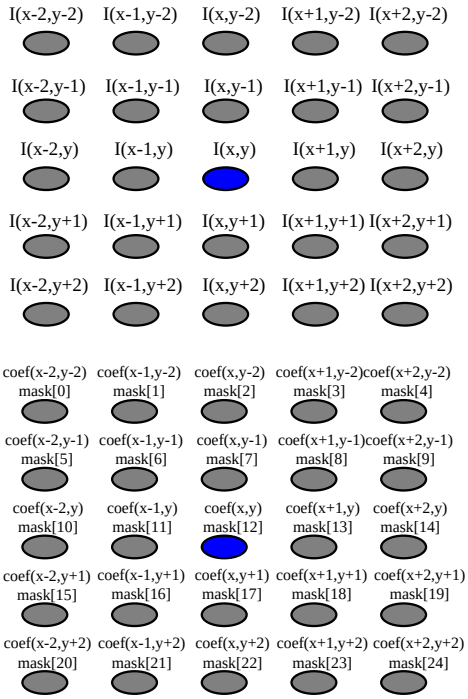
➢ 注意

- 模板系数只能为 0 或255。
- 模板样例

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 255 & 255 & 255 & 0 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 0 & 255 & 255 & 255 & 0 \end{bmatrix} \begin{bmatrix} 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

图2-6 Dilate 计算公式示意图



$$I_{out}(x,y) = \underset{-2\le i\le 2}{\mathrm{O}}\left(\underset{-2\le j\le 2}{\mathrm{O}}\left(f(i,j)\right)\right)$$

其中

$$f(i,j) = I(x-i,y-j)\,\&\,coef(x-i,y-j)$$

$$\underset{-2\le k\le 2}{\mathrm{O}}\left(g(k)\right) = g(-2)\,|\,g(-1)\,|\,g(0)\,|\,g(1)\,|\,g(2)$$

其中，公式中|为位或运算，&为位与运算，%为取余运算。$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst，$coef$(mask)对应pstDilateCtrl中的au8Mask[MI_IVE_MASK_SIZE_5X5]。

- 举例

  无。

- 相关主题

  MI_IVE_Erode

  MI_IVE_OrdStatFilter

1.9. MI_IVE_Erode

- 功能

  执行二值图像 5x5 模板腐蚀任务。

- 语法

  MI_S32 MI_IVE_Erode(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_ErodeCtrl_t *pstErodeCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstErodeCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

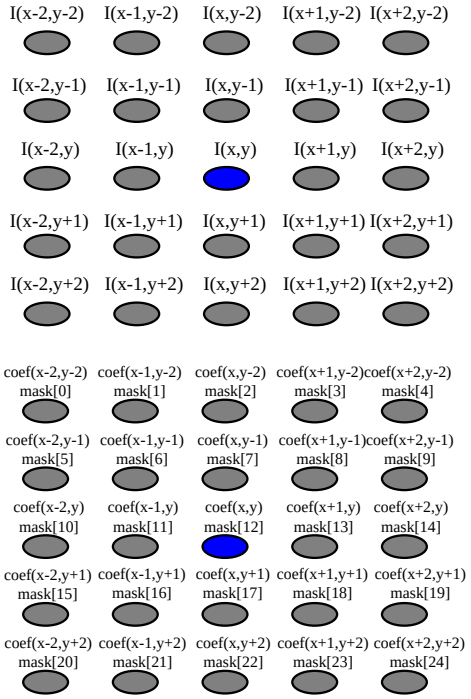
模板系数只能为 0 或255。

模板样例

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 0 & 255 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 255 & 255 & 255 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 255 & 255 & 255 & 0 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 0 & 255 & 255 & 255 & 0 \end{bmatrix} \begin{bmatrix} 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \end{bmatrix}$$

图2-7 Erode 计算公式示意图



$$I_{out}(x,y) = \mathop{\mathrm{O}}_{-2\le i\le 2}\left(\mathop{\mathrm{O}}_{-2\le j\le 2}\left(f(i,j) \right) \right)$$

其中

$$f(i,j)=I(x-i,y-j)\,|\,\big(255-coef\,(x-i,y-j)\big)$$
$$\underset{-2\le k\le 2}{O}\big(g(k)\big)=g(-2)\,\&\,g(-1)\,\&\,g(0)\,\&\,g(1)\,\&\,g(2),$$

其中，公式中|为位或运算，&为位与运算，%为取余运算。 I（x，y）对应 pstSrc，I_{out}（x，y）对应 pstDst， $coef$（mask）对应 pstErodeCtrl 中的 au8Mask[MI_IVE_MASK_SIZE_5X5]。

- 举例

无。

- 相关主题

MI_IVE_Dilate

MI_IVE_OrdStatFilter

1.10. MI_IVE_Thresh

- 功能

执行灰度图像阈值化任务。

- 语法

MI_S32 MI_IVE_Thresh(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_ThreshCtrl_t *pstThrCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstThrCtrl	控制信息指针。	输入
bInstant	保留。	输入

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

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

可以配置 8 种运算模式，具体参见 MI_IVE_ThreshMode_e。

计算公式

E_MI_IVE_THRESH_MODE_BINARY：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & \text{I(x , y)} \le \text{lowThr}\\ \text{maxVal} & \text{I(x , y)} > \text{lowThr}\end{cases}$$

midVal 、highThr 无需赋值。

E_MI_IVE_THRESH_MODE_TRUNC：

$$I_{out}(x,y)=\begin{cases}\text{I(x, y)} & \text{I(x , y)} \le \text{lowThr}\\ \text{maxVal} & \text{I(x , y)} > \text{lowThr}\end{cases}$$

minVal 、midVal 、highThr 无需赋值。

E_MI_IVE_THRESH_MODE_TO_MINVAL：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & \text{I(x , y)} \le \text{lowThr}\\ \text{I(x , y)} & \text{I(x , y)} > \text{lowThr}\end{cases}$$

midVal 、maxVal 、highThr 无需赋值。

E_MI_IVE_THRESH_MODE_MIN_MID_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & \text{I(x , y)} \le \text{lowThr}\\ \text{midVal} & \text{lowThr} \le \text{I(x , y)} \le \text{highThr}\\ \text{maxVal} & \text{I(x , y)} > \text{highThr}\end{cases}$$

E_MI_IVE_THRESH_MODE_ORI_MID_MAX：

$$I_{out}(x,y)=\begin{cases}\text{I(x , y)} & \text{I(x , y)} \le \text{lowThr}\\ \text{midVal} & \text{lowThr} \le \text{I(x , y)} \le \text{highThr}\\ \text{maxVal} & \text{I(x , y)} > \text{highThr}\end{cases}$$

minVal 无需赋值。

E_MI_IVE_THRESH_MODE_MIN_MID_ORI:

$$I_{out}(x,y)=\begin{cases}\text{minVal} & \text{I(x , y)} \le \text{lowThr} \\ \text{midVal} & \text{lowThr} \le \text{I(x , y)} \le \text{highThr} \\ \text{I(x , y)} & \text{I(x , y)} > \text{highThr}\end{cases}$$

maxVal 无需赋值。

E_MI_IVE_THRESH_MODE_MIN_ORI_MAX:

$$I_{out}(x,y)=\begin{cases}\text{minVal} & \text{I(x , y)} \le \text{lowThr} \\ \text{I(x , y)} & \text{lowThr} \le \text{I(x , y)} \le \text{highThr} \\ \text{maxVal} & \text{I(x , y)} > \text{highThr}\end{cases}$$

midVal 无需赋值。

E_MI_IVE_THRESH_MODE_ORI_MID_ORI:

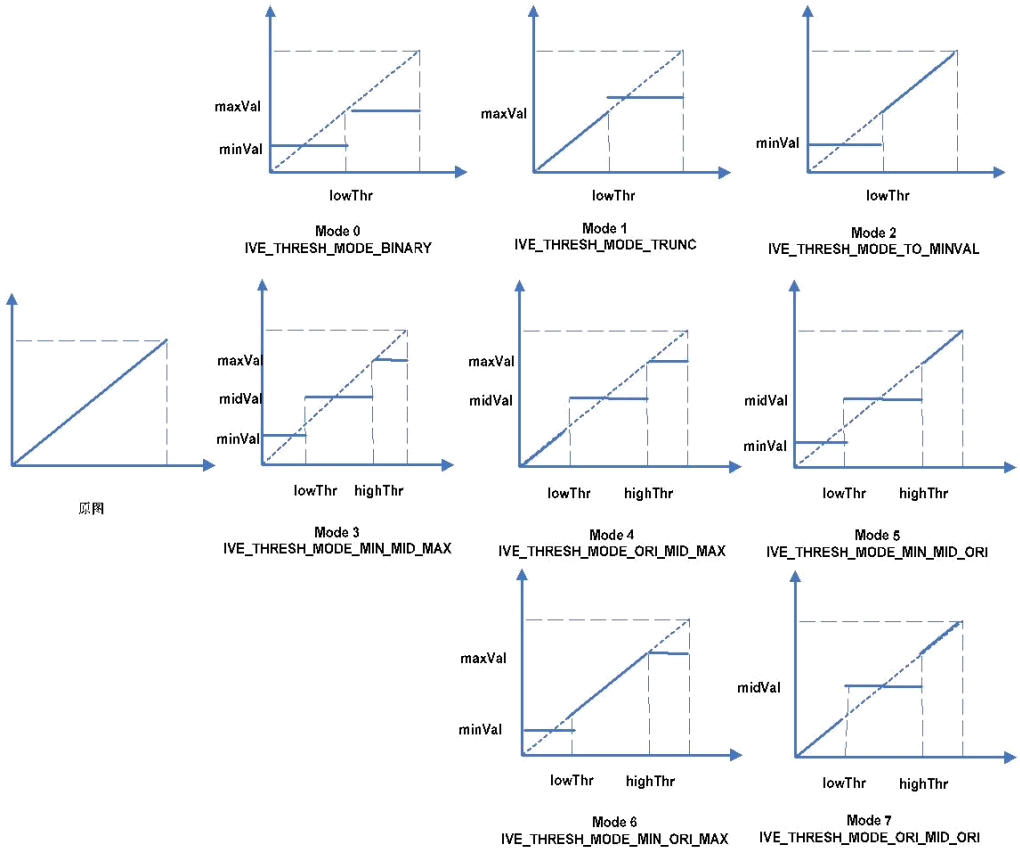
$$I_{out}(x,y)=\begin{cases}\text{I(x , y)} & \text{I(x , y)} \le \text{lowThr} \\ \text{midVal} & \text{lowThr} \le \text{I(x , y)} \le \text{highThr} \\ \text{I(x , y)} & \text{I(x , y)} > \text{highThr}\end{cases}$$

minVal 、maxVal 无需赋值

其中，I(x,y)对应pstSrc，Iout(x,y)对应pstDst，mode、lowThr、highThr、minVal、midVal和maxVal分别对应pstThrCtrl的eMode、u8LowThr、u8HighThr、u8MinVal、u8MidVal和u8MaxVal。具体示意图如图 2-8 所示。

pstThrCtrl 中的 u8MinVal、u8MidVal 和 u8MaxVal 并不需要满足变量命名含义中的大小关系。

图2-8 Thresh 8 种阈值化模式示意图



- 举例

  无。

- 相关主题

  MI_IVE_ThreshS16

  MI_IVE_ThreshU16

1.11. MI_IVE_And

- 功能

  执行两二值图像相与任务。

- 语法

  MI_S32 MI_IVE_And(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	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。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc1。	输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1 的二值图	1 byte	64x64～1920x1080
pstSrc2	U8C1 的二值图	1 byte	同 pstSrc1
pstDst	U8C1 的二值图	1 byte	同 pstSrc1

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

计算公式如下：

$$\mathbf{I}_{out}(x, y) = \mathbf{I}_{src1}(x, y) \,\&\, \mathbf{I}_{src2}(x, y)$$

其中，$(,)1\ I\ x\ y\ src$ 对应 pstSrc1，$(,)2\ I\ x\ y\ src$ 对应 pstSrc2，$I(x, y)\ out$ 对应pstDst

➢ 举例

无。

➢ 相关主题

MI_IVE_Or

MI_IVE_Xor

1.12. MI_IVE_Sub

- 功能

执行两灰度图像相减任务。

- 语法

MI_S32 MI_IVE_Sub(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_IVE_SubCtrl_t *pstSubCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc1。	输出
pstSubCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	64x64～1920x1080
pstSrc2	U8C1	1 byte	同 pstSrc1
pstDst	U8C1、S8C1	1 byte	同 pstSrc1

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

- 依赖

头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h

库文件：libive.a

※ 注意

可以配置 2 种输出格式，具体参见 MI_IVE_SubMode_e。

E_MI_IVE_SUB_MODE_ABS

- 计算公式：$I_{out}(x,y)=abs(I_{src1}(x,y)I_{src2}(x,y))$
- 输出格式：U8C1

E_MI_IVE_SUB_MODE_SHIFT

- 计算公式：$I_{out}(x,y)=(I_{src1}(x,y)I_{src2}(x,y))>>1$
- 输出格式：S8C1

其中，$I_{src1}(x,y)$对应 pstSrc1，$I_{src2}(x,y)$对应 pstSrc2，$I_{out}(x,y)$对应 pstDst。

➢ 举例

无。

➢ 相关主题

MI_IVE_Add

1.13. MI_IVE_Or

➢ 功能

执行两二值图像相或任务。

➢ 语法

MI_S32 MI_IVE_Or(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc1。	输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	64x64～1920x1080
pstSrc2	U8C1	1 byte	同 pstSrc1
pstDst	U8C1	1 byte	同 pstSrc1

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

计算公式如下：

$I_{out}(x,y)=I_{src1}(x,y)\mid I_{src2}(x,y)$

其中，$I_{src1}(x,y)$对应pstSrc1，$I_{src2}(x,y)$对应pstSrc2，$I_{out}(x,y)$对应pstDst。

➢ 举例

无。

➢ 相关主题

MI_IVE_And

MI_IVE_Xor

1.14. MI_IVE_Integ

➢ 功能

执行灰度图像的积分图计算任务。

➢ 语法

MI_S32 MI_IVE_Integ(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_IntegCtrl_t *pstIntegCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstIntegCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	16 byte	32x16～1920x1080
pstDst	U32C1、U64C1	16 byte	同 pstSrc

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

E_MI_IVE_INTEG_OUT_CTRL_COMBINE，组合输出模式，输出图像类型必须为 E_MI_IVE_IMAGE_TYPE_U64C1，参见图 1-13，计算公式如下：

$$I_{sum}(x,y)=\sum_{i\geq 0}^{i\leq x}\sum_{j\geq 0}^{j\leq y} I(i,j)$$

$$I_{sq}(x,y)=\sum_{i\geq 0}^{i\leq x}\sum_{j\geq 0}^{j\leq y}(I(i,j)\bullet I(i,j))$$

$$I_{out}(x,y)=(I_{sq}(x,y)<<28)\,|\,(I_{sum}(x,y)\,\&\,0xFFFFFFF)$$

E_MI_IVE_INTEG_OUT_CTRL_SUM，仅和积分图输出模式，输出图像类型必须为 E_MI_IVE_IMAGE_TYPE_U32C1，计算公式如下：

$$I_{sum}(x,y)=\sum_{i\geq 0}^{i\leq x}\sum_{j\geq 0}^{j\leq y}I(i,j)$$

$$I_{out}(x,y)=I_{sum}(x,y)$$

E_MI_IVE_INTEG_OUT_CTRL_SQSUM，仅平方和积分图输出，输出图像类型必须为E_MI_IVE_IMAGE_TYPE_U64C1，计算公式如下：

$$I_{sq}(x,y)=\sum_{i\geq 0}^{i\leq x}\sum_{j\geq 0}^{j\leq y}(I(i,j)\bullet I(i,j))$$

$$I_{out}(x,y)=I_{sq}(x,y)$$

其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst。

- 举例

无。

- 相关主题

无。

1.15. MI_IVE_Hist

- 功能

执行灰度图像的直方图统计任务。

- 语法

MI_S32 MI_IVE_Hist(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstMemInfo_t *pstDst, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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。 内存至少配置 1024 字节，如图 1-14；	输出
bInstant	保留。	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，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

计算公式如下：

$$I_{out}(x) = \sum_{i}\sum_{j}((I(i,j) == x)?1:0) \quad x = 0...255$$

其中，$I(i,j)$对应pstSrc，$I_{out}(x)$对应pstDst。

※ 注意

无。

➢ 相关主题

无。

1.16. MI_IVE_ThreshS16

➢ 功能

执行 S16 数据到 8bit 数据的阈值化任务。

➢ 语法

MI_S32 MI_IVE_ThreshS16(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_ThreshS16Ctrl_t *pstThrS16Ctrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，RGN_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstThrS16Ctrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

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

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h

库文件：libive. a

※ 注意

可配置 4 种运算模式，参考 MI_IVE_ThreshS16Mode_e。

计算公式

- E_MI_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I(x , y)} \le \text{lowThr})\\ \text{midVal} & (\text{lowThr} < \text{I(x , y)} \le \text{highThr})\\ \text{maxVal} & (\text{I(x , y)} > \text{highThr})\end{cases}$$

要求：-32768≤*lowThr* ≤ *highThr* ≤32767；

-128≤ *minVal* 、*midVal* 、*maxVal* ≤127。

- E_MI_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I}(x,y)\le \text{lowThr})\\ \text{I}(x,y) & (\text{lowThr}<\text{I}(x,y)\le \text{highThr})\\ \text{maxVal} & (\text{I}(x,y)>\text{highThr})\end{cases}$$

要求：-129≤*lowThr* ≤ *highThr* ≤127；

-128≤ *minVal* 、*maxVal* ≤127；

- E_MI_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I}(x,y)\le \text{lowThr})\\ \text{midVal} & (\text{lowThr}<\text{I}(x,y)\le \text{highThr})\\ \text{maxVal} & (\text{I}(x,y)>\text{highThr})\end{cases}$$

要求：-32768≤*lowThr* ≤ *highThr* ≤32767；

0≤ *minVal* 、*midVal* 、*maxVal* ≤255。

- E_MI_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I}(x,y)\le \text{lowThr})\\ \text{I}(x,y) & (\text{lowThr}<\text{I}(x,y)\le \text{highThr})\\ \text{maxVal} & (\text{I}(x,y)>\text{highThr})\end{cases}$$

要求：-1≤*lowThr* ≤ *highThr* ≤255；

0≤ *minVal* 、*maxVal* ≤255。

其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst，mode、lowThr、highThr、minVal、midVal和maxVal分别对应pstThrS16Ctrl的eMode、s16LowThr、s16HighThr、un8MinVal、un8MidVal和un8MaxVal。具体示意图如图 2-9 所示。

pstThrS16Ctrl中的un8MinVal、un8MidVal和un8MaxVal 并不需要满足变量命名含义中的大小关系。

➢ 举例

无。

➢ 相关主题

MI_IVE_ThreshU16

MI_IVE_16BitTo8Bit

1.17. MI_IVE_ThreshU16

➢ 功能

执行 U16 数据到 U8 数据的阈值化任务。

➢ 语法

MI_S32 MI_IVE_ThreshU16(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_ThreshU16Ctrl_t *pstThrU16Ctrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstThrU16Ctrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

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

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h

库文件：libive.a

※ 注意

可配置 2 种运算模式，参考 MI_IVE_ThreshU16Mode_e。

计算公式

- E_MI_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I(x , y)} \le \text{lowThr})\\ \text{midVal} & (\text{lowThr} < \text{I(x , y)} \le \text{highThr})\\ \text{maxVal} & (\text{I(x , y)} > \text{highThr})\end{cases}$$

要求：0≤ *lowThr* ≤ *highThr* ≤65535；

- E_MI_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX：

$$I_{out}(x,y)=\begin{cases}\text{minVal} & (\text{I(x , y)} \le \text{lowThr})\\ \text{I(x , y)} & (\text{lowThr} < \text{I(x , y)} \le \text{highThr})\\ \text{maxVal} & (\text{I(x , y)} > \text{highThr})\end{cases}$$

要求：0≤ *lowThr* ≤ *highThr* r≤255；

其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst，mode、lowThr、highThr、minVal、midVal 和maxVal分别对应pstThrU16Ctrl的 eMode、u16LowThr、u16HighThr、u8MinVal、u8MidVal和u8MaxVal。具体示意图如图 2-10 所示。

pstThrU16Ctrl 中的u8MinVal、u8MidVal 和 u8MaxVal 并不需要满足变量命名含义中的大小关系。

➢ 举例

无。

➢ 相关主题

MI_IVE_ThreshS16

MI_IVE_16BitTo8Bit

1.18. MI_IVE_16BitTo8Bit

➢ 功能

执行 16bit 图像数据到 8bit 图像数据的线性转化任务。

➢ 语法

MI_S32 MI_IVE_16BitTo8Bit(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_16bitTo8BitCtrl_t *pst16BitTo8BitCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pst16BitTo8BitCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U16C1、S16C1	2 byte	64x64～1920x1080
pstDst	U8C1、S8C1	1 byte	同 pstSrc

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

- 依赖

头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
库文件：libive. a

- 注意

- 可配置 4 种模式，具体参考 MI_IVE_16BitTo8BitMode_e。
  计算公式
  - E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8：

$$I_{out}(x,y)=\begin{cases}-128 & (\frac{a}{b}I(x,y)<-128)\\ \frac{a}{b}I(x,y) & (-128\le\frac{a}{b}I(x,y)\le 127)\\ 127 & (\frac{a}{b}I(x,y)>127)\end{cases}$$

- E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS：

$$I_{out}(x,y)=\begin{cases}\left|\frac{a}{b}I(x,y)\right| & (\left|\frac{a}{b}I(x,y)\right|\le 255)\\ 255 & (\left|\frac{a}{b}I(x,y)\right|>255)\end{cases}$$

- E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS：

$$I_{out}(x,y)=\begin{cases}0 & (\frac{a}{b}I(x,y)+bais<0)\\ \frac{a}{b}I(x,y)+bais & (0\le\frac{a}{b}I(x,y)+bais\le 255)\\ 255 & (\frac{a}{b}I(x,y)+bais>255)\end{cases}$$

- E_MI_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8：

$$I_{out}(x,y)=\begin{cases}0 & (\frac{a}{b}I(x,y)<0)\\ \frac{a}{b}I(x,y) & (0\le\frac{a}{b}I(x,y)\le 255)\\ 255 & (\frac{a}{b}I(x,y)>255)\end{cases}$$

其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst，mode、 a、 b和 $bias$分别对应pst16BitTo8BitCtrl的eMode、u8Numerator、u16Denominator、s8Bias。具体示意图如图 2-11 所示。

要求：u8Numerator ≤ u16Denominator，且 u16Denominator≠0。

➢ 相关主题

MI_IVE_ThreshS16

MI_IVE_ThreshU16

1.19. MI_IVE_OrdStatFilter

➢ 功能

执行 3x3 模板顺序统计量滤波任务，可进行 Median、Max、Min 滤波。

➢ 语法

MI_S32 MI_IVE_OrdStatFilter(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_OrdStatFilter_t *pstOrdStatFltCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstOrdStatFltCtrl	控制参数指针不能为空。	输入
bInstant	保留。	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

- 可配置 3 种滤波模式，参考 MI_IVE_OrdStatFilterMode_e。
  计算公式
  - E_MI_IVE_ORD_STAT_FILTER_MODE_MEDIAN：
    $$I_{out}(x,y) = \underset{-1\le i\le 1,-1\le j\le 1}{\text{median}}\{I(x+i,y+j)\}$$
  - E_MI_IVE_ORD_STAT_FILTER_MODE_MAX：
    $$I_{out}(x,y) = \underset{-1\le i\le 1,-1\le j\le 1}{\max}\{I(x+i,y+j)\}$$
  - E_MI_IVE_ORD_STAT_FILTER_MODE_MIN：
    $$I_{out}(x,y) = \underset{-1\le i\le 1,-1\le j\le 1}{\min}\{I(x+i,y+j)\}$$

  其中，$I(x,y)$对应pstSrc，$I_{out}(x,y)$对应pstDst。

- 举例

  无。

- 相关主题

  MI_IVE_Filter

  MI_IVE_Dilate

  MI_IVE_Erode

1.20. MI_IVE_Map

- 功能

  执行 Map（映射赋值）任务，对源图像中的每个像素，查找 Map 查找表中的值，赋予目标图像相应像素查找表中的值，支持 U8C1→U8C1模式的映像。

- 语法

  MI_S32 MI_IVE_Map(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_SrcMemInfo_t *pstMap, MI_IVE_DstImage_t *pstDst, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstMap	映射表信息指针。不能为空。 内存至少配置：sizeof(MI_IVE_MapLutMem_t)。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1	1 byte	64x64～1920x1080
pstMap	-	16 byte	-
pstDst	U8C1	1 byte	同 pstSrc

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

计算公式如下：

$$I_{out}(x, y) = map[I(x, y)]$$

其中，$I(x, y)$对应pstSrc，$I_{out}(x, y)$对应pstDst，map对应pstMap。

➢ 举例

无。

➢ 相关主题

无。

1.21. MI_IVE_EqualizeHist

➢ 功能

执行灰度图像的直方图均衡化计算任务。

➢ 语法

MI_S32 MI_IVE_EqualizeHist(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MI_IVE_EqualizeHistCtrl_t *pstEqualizeHistCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出

参数名称	描述	输入/输出
pstEqualizeHistCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	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
pstEqualizeHistCtrl→stMem	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

- pstEqualizeHistCtrl 中的stMem，至少需开辟 sizeof(MI_IVE_EqualizeHistCtrlMem_t)字节大小。
- 与OpenCV 中直方图均衡化计算过程一致。

➢ 举例

无。

➢ 相关主题

无。

1.22. MI_IVE_Add

➢ 功能

执行两灰度图像的加权加计算任务。

➢ 语法

MI_S32 MI_IVE_Add(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_IVE_AddCtrl_t *pstAddCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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。	输入
pstDst	输出图像指针。 高、宽同 pstSrc1；不能为空。	输出
pstAddCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	64x64～1920x1080
pstSrc2	U8C1	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

计算公式如下：

$$I_{out}(x,y) = x * I_{src1}(x,y) + y * I_{src2}(x,y)$$

其中，$I_1(i,j)$对应pstSrc1，$I_2(i,j)$对应pstSrc2，$I_{out}(i,j)$对应pstDst；x，y为pstAddCtrl中的u0q16X，u0q16Y；要求定点化前的$0<x<1$，$0<y<1$，且$x+y=1$。

- 举例

  无。

- 相关主题

  MI_IVE_Sub

1.23. MI_IVE_Xor

- 功能

  执行两二值图的异或计算任务。

- 语法

  MI_S32 MI_IVE_Xor(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc1	源图像 1 指针。不能为空。	输入
pstSrc2	源图像 1 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
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	64x64～1920x1080
pstSrc2	U8C1	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

计算公式如下：

$$I_{out}(x,y) = I_{src1}(x,y) \wedge I_{src2}(x,y)$$

其中，$I_{src1}(x,y)$对应pstSrc1，$I_{src_2}(x,y)$对应pstSrc2，$I_{dst}(x,y)$对应pstDst

➢ 举例

无。

➢ 相关主题

- MI_IVE_And
- MI_IVE_Or

1.24. MI_IVE_Ncc

➢ 功能

执行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

MI_S32 MI_IVE_Ncc(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstMemInfo_t *pstDst, MI_BOOL bInstant);

➢ 返回值

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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、宽同 pstSrc1。	输入
pstDst	输出数据指针。不能为空。 内存至少需配置：sizeof（MI_IVE_NccDstMem_t）。	输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	32x32～1920x1080
pstSrc2	U8C1	1 byte	同 pstSrc
pstDst	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

- 计算公式如下

$$NCC(\mathrm{I}_{src1},\mathrm{I}_{src2}) = \frac{\sum_{i=1}^{w}\sum_{j=1}^{h}(I_{src1}(i,j)*I_{src2}(i,j))}{\sqrt{\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src1}(i,j))}\sqrt{\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src2}(i,j))}}$$

- 仅输出上面公式的分子、开方之前的两个分母项，即 pstDst→u64Numerator、pstDst→u64QuadSum1、pstDst→u64QuadSum2 分别对应上面公式的 $\sum_{i=1}^{w}\sum_{j=1}^{h}(I_{src1}(i,j)*I_{src2}(i,j))$ 、 $\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src1}(i,j))$ 、 $\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src2}(i,j))$

➢ 举例

无。

➢ 相关主题

无。

1.25. MI_IVE_Ccl

➢ 功能

执行二值图像的连通区域标记任务。

➢ 语法

MI_S32 MI_IVE_Ccl(MI_IVE_HANDLE hHandle, MI_IVE_Image_t *pstSrcDst, MI_IVE_DstMemInfo_t *pstBlob, MI_IVE_CclCtrl_t *pstCclCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrcDst	源图像指针，连通区域标记在源图像上进行，即源图像同时也是标记图像输出。不能为空。	输入、输出
pstBlob	连通区域信息指针。不能为空。 内存至少需配置为 sizeof（MI_IVE_CcBlob_t)大小，最多输出 254 个有效的连通区域。	输出
pstCclCtrl	控制参数指针不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1	16 byte	-
pstBlob	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

- 连通区域的信息保存在 pstBlob→astRegion 中。
- pstBlob→u8RegionNum 表示有效的连通区域数目，最多 254 个有效的连通区域；有效的连通区域的面积大于 pstBlob→u16CurAreaThr，标记号为其所在 pstBlob→astRegion 数组元素的下标+1。有效的连通区域并不一定连续地存储在数组中，而很可能是间断的分布在数组中。
- 若pstBlob→s8LabelStatus 为0，则标记成功（一个区域一个标记；若为-1，则标记失败（一个区域多个标记或者多个区域共享一个标记）对于后者，若用户需要正确的标记号，还需要再次根据 pstBlob 中的外接矩形信息重新标记。不管标记是否成功，连通区域的外接矩形信息一定是正确可用的。

- 输出的连通区域会用 pstCclCtrl→u16InitAreaThr 进行筛选，面积小于等于 pstCclCtrl→u16InitAreaThr 均会被置为 0。
- 当连通区域数目大于 254，会用 pstCclCtrl→u16InitAreaThr 删除面积小的连通区域；若 pstCclCtrl→u16InitAreaThr 不满足删除条件，会以 pstCclCtrl→u16Step 为步长，增大删除连通区域的面积阈值。
- 最终的面积阈值存储在 pstBlob→u16CurAreaThr 中。另外再用索引254记录被删除的连通区域总面积。

➢ 举例

无。

➢ 相关主题

无。

1.26. MI_IVE_Gmm

➢ 功能

执行 GMM 背景建模任务，支持灰度图、RGB_PACKAGE 图像的 GMM 背景建模，高斯模型个数为 3 或者 5。

➢ 语法

MI_S32 MI_IVE_Gmm(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstFg, MI_IVE_DstImage_t *pstBg, MI_IVE_MemInfo_t *pstModel, MI_IVE_GmmCtrl_t *pstGmmCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstFg	前景图像指针。不能为空。 高、宽同 pstSrc。	输出
pstBg	背景图像指针。不能为空。 高、宽同 pstSrc。	输出
pstModel	GMM 模型参数指针。不能为空。	输入、输出
pstGmmCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc	U8C1、U8C3_PACKAGE	16 byte	-
pstFg	U8C1 的二值图	16 byte	-
pstBg	同 pstSrc	16 byte	-
pstModel	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

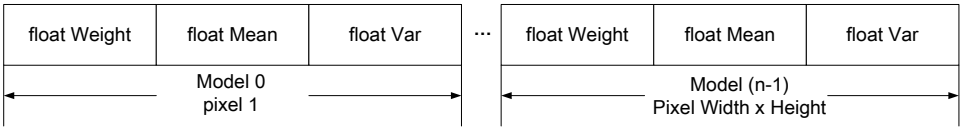
➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

- GMM 的实现方式参考了 OpenCV 中的MOG 和MOG2。
- 源图像类型只能为 U8C1 或U8C3_PACKAGE，分别用于灰度图和 RGB 图的 GMM 背景建模。
- 前景图像是二值图，类型只能为 U8C1；背景图像与源图像类型一致。
- 灰度图像 GMM 采用 n 个(n=3 或 5})高斯模型，pstModel 的内存排列方式如图2-12 所示。

图2-12 灰度图像 GMM 模型的内存配置示意图

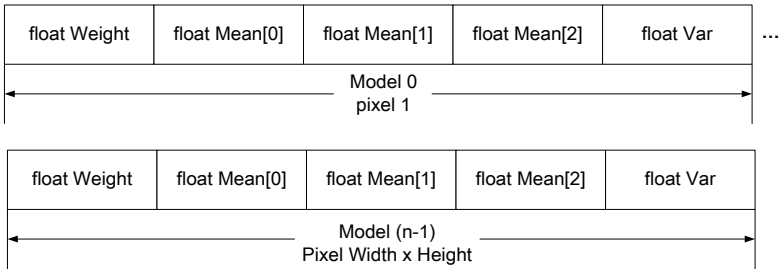


一个像素的单个高斯模型参数 weight 用2 字节、mean 用2 字节、var 用3 字节； 因此pstModel 需要分配的内存大小：

pstModel→u32Size = 7 * pstSrc→u16Width * pstSrc→u16Height * pstGmmCtrl→u8ModeNum

- RGB 图像GMM 采用 n 个(n=3 或 5})高斯模型，pstModel 的内存排列方式如图2-13 所示。

图2-13 RGB 图像 GMM 模型的内存配置示意图



一个像素的单个高斯模型参数 weight 用4 字节、mean[3]用4*3 字节、var 用4 字节； 因此pstModel 需要分配的内存大小：

pstModel→u32Size = 20 * pstSrc→u16Width * pstSrc→u16Height * pstGmmCtrl→u8ModeNum

➢ 举例

无。

➢ 相关主题

无

1.27. MI_IVE_CannyHysEdge

➢ 功能

灰度图的 Canny 边缘提取的前半部：求梯度、计算梯度幅值幅角、磁滞阈值化及非极大抑制。

➢ 语法

MI_S32 MI_IVE_CannyHysEdge(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstEdge, MI_IVE_DstMemInfo_t *pstStack, MI_IVE_CannyHysEdgeCtrl_t *pstCannyHysEdgeCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstEdge	强弱边缘标志图像指针。不能为空。 高、宽同 pstSrc。	输出
pstStack	强边缘点坐标栈。不能为空。 内存至少配置： pstSrc→u16Width * pstSrc→u16Height *	输出

参数名称	描述	输入/输出
	(sizeof(MI_IVE_PointU16_t)) + sizeof(MI_IVE_CannyStackSize_t)	
pstCannyHysEdgeCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	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
pstEdge	U8C1	16 byte	同 pstSrc
pstStack	-	16 byte	-
pstCannyHysEdgeCtrl→stMem	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

- pstEdge 仅有0、1、2 三个取值：
  - 0 表示弱边缘点
  - 1 表示非边缘点
  - 2 表示强边缘点
- pstStack 中存储强边缘点的坐标信息。
- pstCannyHysEdgeCtrl→stMem 至少需要分配的内存大小
  pstCannyHysEdgeCtrl→stMem.u32Size
  = IveGetStride(pstSrc→u16Width, MI_IVE_STRIDE_ALIGN)* 3 * pstSrc→u16Height。
- 该任务完成后，必须要使用 MI_IVE_CannyEdge_函数才能输出 Canny 边缘图像。

➢ 举例

无。

➢ 相关主题

MI_IVE_CannyEdge

1.28. MI_IVE_CannyEdge

➢ 功能

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

MI_S32 MI_IVE_CannyEdge(MI_IVE_HANDLE hHandle, MI_IVE_Image_t *pstEdge, MI_IVE_MemInfo_t *pstStack, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstEdge	作为输入是强弱边缘标志图像指针；作为输出是边缘二值图像指针。 不能为空。	输入、输出
pstStack	强边缘点坐标栈。不能为空。	输入、输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstEdge	U8C1	16 byte	64x64～1920x1024
pstStack	-	16 byte	-

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

使用该接口前必须调用 MI_IVE_CannyHysEdge，在保证 MI_IVE_CannyHysEdge_任务完成的情况下，使用 MI_IVE_CannyHysEdge_的输出 pstEdge、pstStack 作为该接口的参数输入。

- 举例

  无。

- 相关主题

  MI_IVE_CannyHysEdge

1.29. MI_IVE_Lbp

- 功能

  执行 LBP 计算任务。

- 语法

  MI_S32 MI_IVE_Lbp(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_DstImage_t *pstDst, MI_IVE_LbpCtrrl_t *pstLbpCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc1	源图像指针。不能为空。	输入
pstSrc2	源图像指针。不能为空。 如果输入的 channel mode 是 U8C1.则可以为空。	输入
pstDst	输出图像指针。不能为空。 高、宽同 pstSrc。	输出
pstLbpCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	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	64x64~1920x1024

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

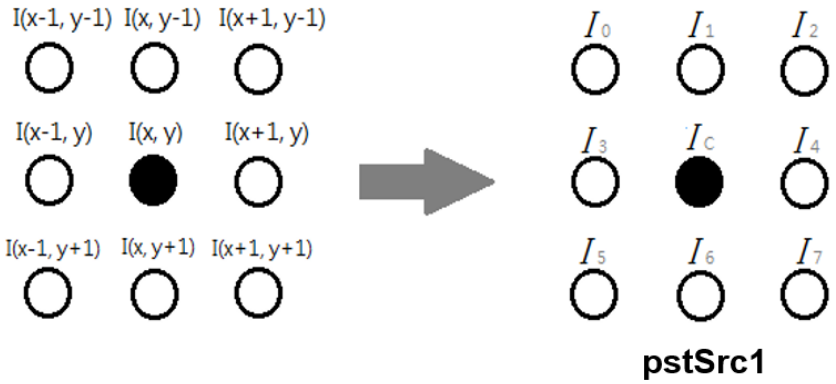
➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

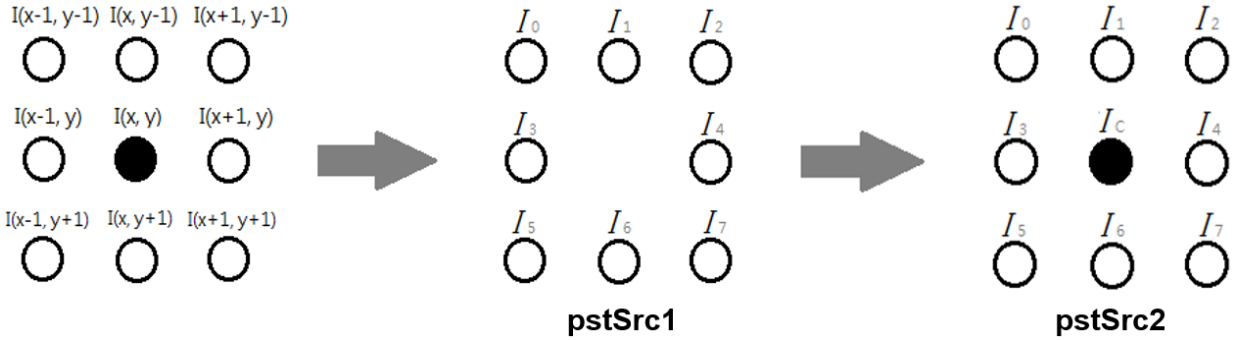
※ 注意

LBP U8C1 mode的计算公式如图 2-16 所示。

图2-16 LBP 计算公式示意图



LBP U8C2 mode的计算示意图如下



- E_MI_IVE_LBP_CMP_NORMAL

$$lbp(x,y)=\sum_{i=0}^{7}((I_i-I_c)>=thr)<<(7-i),thr\in[-128,127]$$

- E_MI_IVE_LBP_CMP_ABS

$$lbp(x,y)=\sum_{i=0}^{7}(abs(I_i-I_c)>=thr)<<(7-i),thr\in[0,255]$$

- E_MI_IVE_LBP_CMP_ABS_MUL

$$lbp(x,y)=\sum_{i=0}^{7}\left(abs\left(I_i-I_c\right)\geq thr\times I_c\right)\ll(7-i),thr\in[0,1]$$

其中，在 U8C1 模式下，$I(x,y)$对应pstSrc1，$lpb(x,y)$对应pstDst，thr对应pstLbpCtrl→un8BitThr。而U8C2 模式下，I_c 对应的是 pstSrc2

➢ 举例

无。

➢ 相关主题

无。

1.30. MI_IVE_NormGrad

➢ 功能

执行归一化梯度计算任务，梯度分量均归一化到 S8。

➢ 语法

MI_S32 MI_IVE_NormGrad(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDstH, MI_IVE_DstImage_t *pstDstV, MI_IVE_DstImage_t *pstDstHV, MI_IVE_NormGradCtrl_t *pstNormGradCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDstH	由模板直接滤波并归一到 S8 后得到的梯度分量图像(H)指针。 根据 pstNormGradCtrl→eOutCtrl，若需要输出则不能为空。	输出
pstDstV	由转置后的模板滤波并归一到 S8 后得到的梯度分量图像(V)指针。 根据 pstNormGradCtrl→eOutCtrl，若需要输出则不能为空。	输出
pstDstHV	由模板和转置后的模板直接滤波，并且均归一到 S8 后，采用package 格式存储（如图 1-7）的图像指针。 根据 pstNormGradCtrl→eOutCtrl，若需要输出则不能为空。	输出
pstNormGradCtrl	控制信息指针。	输入
bInstant	保留。	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
pstDstH	S8C1	16 byte	同 pstSrc
pstDstV	S8C1	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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

- 控制参数中输出模式如下：
  - E_MI_IVE_NORM_GRAD_OUT_CTRL_HOR_AND_VER 时，pstDstH 和pstDstV 指针不能为空，且要求跨度一致；
  - E_MI_IVE_NORM_GRAD_OUT_CTRL_HOR 时，pstDstH 不能为空；
  - E_MI_IVE_NORM_GRAD_OUT_CTRL_VER 时，pstDstV 不能为空；
  - E_MI_IVE_NORM_GRAD_OUT_CTRL_COMBINE 时，pstDstHV 不能为空。

➢ 举例

无。

➢ 相关主题

MI_IVE_Sobel

1.31. MI_IVE_LkOpticalFlow

➢ 功能

执行单层 LK 光流计算任务。

➢ 语法

MI_S32 MI_IVE_LKOpticalFlow(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrcPre, MI_IVE_SrcImage_t *pstSrcCur, MI_IVE_SrcMemInfo_t *pstPoint, MI_IVE_MemInfo_t *pstMv, MI_IVE_LkOpticalFlowCtrl_t *pstLkOptiFlowCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrcPre	前一帧图像指针。不能为空。	输入
pstSrcCur	当前图像指针。不能为空。 高、宽同 pstSrcPre。	输入
pstPoint	当前金字塔层的初始特征点坐标。不能为空。 坐标只能为 MI_IVE_PointS25Q7_t 类型；内存至少需分配：pstLkOptiFlowCtrl→u16CornerNum * sizeof(MI_IVE_PointS25Q7_t)。	输入
pstMv	对应于 pstPoint 的特征点运动位移矢量。不能为空。 首次计算需初始化为 0 输入；后续层计算需输入上一层计算得到的运动位移矢量；位移只能为 MI_IVE_MvS9Q7_t 类型；内存至少需分配：pstLkOptiFlowCtrl→u16CornerNum * sizeof(MI_IVE_MvS9Q7_t)	输入、输出
pstLkOptiFlowCtrl	控制参数指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcPre	U8C1	16 byte	64x64～720x576
pstSrcCur	U8C1	16 byte	同 pstSrcPre

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

- 依赖

  - 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
  - 库文件：libive. a

➢ 注意

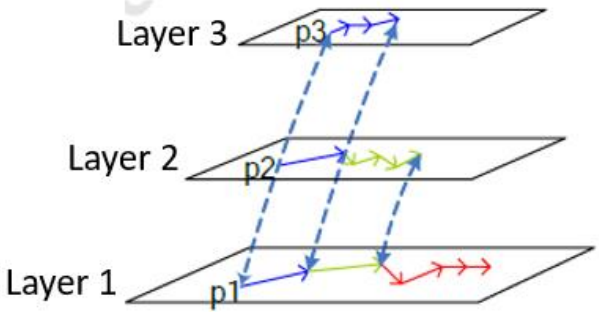
- 求解下面的光流方程中，仅用到特征点周围 7X7 像素的来计算对应的I_x、I_y、I_t

$$\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum I_x I_t \\ -\sum I_y I_t \end{bmatrix}$$

其中，I_x、I_y、I_t分别表示当前图像在x、y方向的偏导，当前图像与前一帧图像的差分。

- 以3 层金字塔 LK 光流计算为例，要求每层图像的高、宽是上一层图像高、宽的一半，其计算示意图如图2-18 所示。

图2-18 3 层金字塔 LK 光流计算示意图



- 根据输入的特征点坐标，计算出 3 层金字塔特征点对应的坐标：p0，p1，p2；
- 以p2 和初始为 0 的mv2 作为输入调用 LK 算子求出在第 2 层上的位移 mv2；
- 以p1 和mv2 作为输入调用 LK 算子求出第 1 层上的位移 mv1；
- 以p0 和mv1 作为输入调用 LK 算子求出第 0 层上的位移 mv0；
- 若第 0 层不是原始图像，根据第 0 层与原始图像的的比例关系可以得到 LK 光流的真正位移 mv。

请注意设计和使用限制：每个特征点仅以该特征点为中心固定大小窗口的数据进 行计算，若迭代计算过程中，该特征点位移目标点超出该固定大小窗口会导致计 算光流失败。

➢ 举例

无。

➢ 相关主题

无。

1.32. MI_IVE_Sad

➢ 功能

计算两幅图像按 4x4\8x8\16x16 分块的 16 bit\8 bit SAD 图像，以及对 SAD 进行阈值化输出。

- 语法

MI_S32 MI_IVE_Sad(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstSad, MI_IVE_DstImage_t *pstThr, MI_IVE_SadCtrl_t *pstSadCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	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。	输入
pstSad	输出 SAD 图像指针。 根据 pstSadCtrl→eOutCtrl，若需要输出则不能为空。 根据 pstSadCtrl→eMode，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 pstSadCtrl→eOutCtrl，若需要输出则不能为空。 根据 pstSadCtrl→eMode，对应 4x4、8x8、16x16 分块模式，高、宽分别为 pstSrc1 的 1/4、1/8、1/16。	输出
pstSadCtrl	控制信息指针。 不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	U8C1	1 byte	64x64～1920x1080
pstSrc2	U8C1	1 byte	同 pstSrc1
pstSad	U8C1 、U16C1	16 byte	根据 pstSadCtrl→eMode， 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→eMode， 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，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

计算公式如下：

$$SAD_{out}(x,y) = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} abs(I_{src1}(M*x+i, M*y+j) - I_{src2}(M*x+i, M*y+j))$$

$$\mathrm{Thr}(x,y) = \begin{cases} minVal \cdots O(x,y) \le Thresh \\ maxVal \cdots O(x,y) > Thresh \end{cases}$$

其中，$I_1(i,j)$对应pstSrc1，$I_2(i,j)$对应pstSrc2，$SAD_{out}(x,y)$对应pstSad，n与pstSadCtrl→eMode相关，对应E_MI_IVE_SAD_MODE_MB_4X4、E_MI_IVE_SAD_MODE_MB_8X8、E_MI_IVE_SAD_MODE_MB_16X16 时分别取 4、8、16；$THR_{out}(x,y)$对应pstThr，*Thr*、*minVal*和*maxVal*分别对应pstSadCtrl→u16Thr、pstSadCtrl→u8MinVal和pstSadCtrl→u8MaxVal。

➢ 举例

无。

➢ 相关主题

无。

1.33. MI_IVE_Bernsen

➢ 功能

执行 3x3 和 5x5 模板之 Bernsen 门坎值任务。

➢ 语法

MI_S32 MI_IVE_Bernsen(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst,MVE_IVE_BernsenCtrl_t *pstBernsenCtrl,MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0, MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出 Bernsen 图像指针。	输出
pstBernsenCtrl	控制信息指针。 不能为空。	输入
bInstant	保留。	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	S8C1	16 byte	同 pstSrc

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

※ 注意

目前支持两种模式，计算公式如下：

- MVE_BERNSEN_MODE_NORMAL

$$T(x,y)=0.5\times\left(\max_{-\omega\le i\le\omega,-\omega\le j\le\omega} I(x+i,y+j)+\min_{-\omega\le i\le\omega,-\omega\le j\le\omega} I(x+i,y+j)\right)$$

$$I_{out}(x,y)=\begin{cases}0 & I(x,y)<T(x,y)\\ 1 & I(x,y)\ge T(x,y)\end{cases}$$

pstBernsenCtrl->u8Thr 无须设定

- MVE_BERNSEN_MODE_THRESH

$$T(x,y)=0.5\times\left(\max_{\substack{-\omega\le i\le\omega\\-\omega\le j\le\omega}} I(x+i,y+j)+\min_{\substack{-\omega\le i\le\omega\\-\omega\le j\le\omega}} I(x+i,y+j)\right)$$

$$I_{out}(x,y)=\begin{cases}0 & I(x,y)<0.5\times(T(x,y)+Thr)\\ 1 & I(x,y)\ge 0.5\times(T(x,y)+Thr)\end{cases}$$

I(x,y) 对应到 pstSrc, Iout(x,y) 对应到 pstDst, and Thr 对应到 pstBernsenCtrl 里的 u8thr.

➢ 举例

无。

➢ 相关主题

无。

1.34. MI_IVE_LineFilterHor

➢ 功能

针对二位图像进行水平方向的滤波任务。

➢ 语法

MI_S32 MI_IVE_LineFilterHor(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrcDst, MVE_IVE_LineFilterHorCtrl_t *pstLineFilterHorCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrcDst	源图像与输出图像指针。不能为空。	输入/输出
pstLineFilterHorCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1 的二值化影像	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

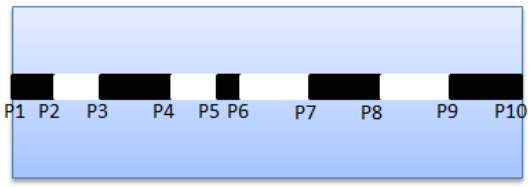
- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

对输入图像进行水平方向扫描，当发现黑色线段的长度满足下面条件所述，则将此黑色线段转换为白色线段。

计算公式如下：

步骤一 ：先水平扫描，并记录水平黑线和白点交错信息，范例如下



步骤二 ：针对上述步骤进行水平扫描黑转白操作。

举例 ：假设当前黑色线段为 line_black56，如果此线段长度满足下面条件，及转换为白线

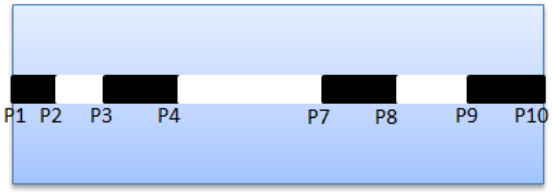
条件 1：line_black34 > thr1

条件 2：line_black78 > thr1

条件 3：

(line_white45 + line_black56 + line_white67) <= (line_white45 + line_white67)×thr2

条件 4：(line_white45 + line_white67) > 1



步骤三 ：继续上述二步骤，直到处理完整张影像，结果如下

步骤四 ：根据上面结果重新进行第二次扫描，并根据黑线与白线的关系进行黑转白操作。

举例 ：假设当前黑线为line_black34，如果满足下面条件，则将黑线转为白线。

条件 1：line_block34 < thr3

条件 2：line_white47 > 2×thr3

条件 3：line_white23 > 9

条件 4：line_white23 < 3×thr3

步骤五 ：反复执行部步骤三和四，直到整张图像处理完毕。

➢ 举例

无。

1.35. MI_IVE_LineFilterVer

➢ 功能

针对二位图像进行垂直方向的滤波任务。

➢ 语法

MI_S32 MI_IVE_LineFilterVer(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrcDst, MVE_IVE_LineFilterVerCtrl_t *pstLineFilterVerCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrcDst	源图像与输出图像指针。不能为空。	输入/输出
pstLineFilterVerCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1 的二值化影像	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

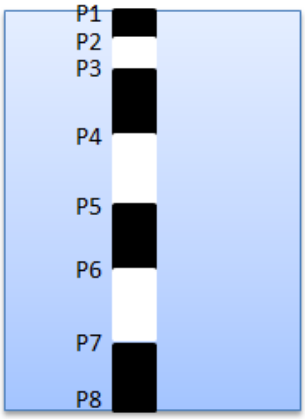
- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

对输入图像进行垂直方向扫描，当发现黑色线段的长度满足下面条件所述，则将此黑色线段转换为白色线段。

计算公式如下

步骤一 ：先垂直扫描，并记录水平黑线和白点交错信息，范例如下



步骤二：假设当前黑色线段为 line_black56，若黑色线段长度满足下面条件，则将黑线转白线。

Condition 1 : line_black56 < thr

Condition 2 : line_black67 > 6 & line_whitr67 < 25

Condition 3 : line_white45 < 12

步骤三 ： 根据前面步骤一和步骤二，将整张图像处理完毕。

- 举例

无。

1.36. MI_IVE_NoiseRemoveHor

- 功能

针对二位图像进行水平方向之噪声滤除任务。

- 语法

MI_S32 MI_IVE_NoiseRemoveHor(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrcDst, MVE_IVE_NoiseRemoveHorCtrl_t *pstNoiseRemoveHorCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrcDst	源图像与输出图像指针。不能为空。	输入/输出
pstNoiseRemoveHorCtrl	控制信 息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1 的二值化影像	16 byte	64x64～1920x1080

- 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

对输入图像进行水平方向扫描，当白色线段长度小于 thr1，或是大于 thr2，则将白线转化为黑线。

➢ 举例

无。

1.37. MI_IVE_NoiseRemoveVer

➢ 功能

针对二位图像进行水平方向之噪声滤除任务。

➢ 语法

MI_S32 MI_IVE_NoiseRemoveHor(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrcDst, MVE_IVE_NoiseRemoveVerCtrl_t *pstNoiseRemoveVerCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrcDst	源图像与输出图像指针。不能为空。	输入/输出
pstNoiseRemoveVerCtrl	控制信 息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1 的二值化影像	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

对输入图像进行垂直方向扫描，当白色线段长度小于 thr1，或是大于 thr2，则将白线转化为黑线。

➢ 举例

无。

1.38. MI_IVE_AdpThresh

➢ 功能

执行使用自适应性门坎值的二值化任务。

➢ 语法

MI_S32 MI_IVE_AdpThresh(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_SrcImage_t *pstInteg, MI_IVE_DstImage_t *pstDst, MVE_IVE_AdpThreshCtrl_t *pstAdpThrCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstInteg	输入积分影像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。	输出
pstAdpThrCtrl	控制信 息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1	16 byte	64x64～1920x1080
pstInteg	U32C1	16 byte	64x64～1920x1080
pstDst	U8C1	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

计算公式如下

$$T(x,y)=\frac{1}{w\times h}\times\sum_{i=-h/2}^{h/2}\sum_{j=-w/2}^{w/2}I(x+i,y+j)$$

$$w = u8HalfMaskx$$

$$h = u8HalfMasky$$

$$RateThr = u8RateThr/10$$

$$Offset = s16Offset$$

$$ValueThr = u8ValueThr$$

$$I_{out}(x,y)=\begin{cases}0 & I(x,y)\le(T(x,y)\times RateThr)-Offset \ \&\ I(x,y)<ValueThr\\ 1 & I(x,y)>(T(x,y)\times RateThr)-Offset \ \|\ I(x,y)\ge ValueThr\end{cases}$$

I(x, y) 对应到 pstSrc, Iout(x,y) 对应到 pstDst, W , h , RateThr , 0ffset 和 ValueThr 对应到 pstAdpThrCtrl 里的 u8HalfMaskx , u8HalfMasky , u8RateThr , s16Offset 和 u8ValueThr. 积分影像计算公式如下

$$\sum_{i=-h/2}^{h/2}\sum_{j=-w/2}^{w/2}I(x+i,y+j)$$

➢ 举例

无。

1.39. MI_IVE_Resize

➢ 功能

执行缩放影像任务。

➢ 语法

MI_S32 MI_IVE_Resize(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstImage_t *pstDst, MVE_IVE_ResizeCtrl_t * pstResizeCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDst	输出图像指针。不能为空。	输出
pstResizeCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrcDst	U8C1, U8C3_PLANAR, U8C3_PACKAGE 和 YUV420SP	16 byte	64x64～1920x1080
pstDst	U8C1, U8C3_PLANAR, U8C3_PACKAGE and YUV420SP	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

※ 注意

实现方法参考标准的 bilinear。

➢ 举例

无。

1.40. MI_IVE_Bat

➢ 功能

针对二位图像执行水平和垂直方向的数值交替次数计算任务。

➢ 语法

MI_S32 MI_IVE_BAT (MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc, MI_IVE_DstMemInfo_t *pstDstH, MI_IVE_DstMemInfo_t *pstDstV, MVE_IVE_BatCtrl_t *pstCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc	源图像指针。不能为空。	输入
pstDstH	输t出水平图像指针。不能为空。	输出
pstDstV	输t出垂直图像指针。不能为空。	输出
pstCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

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

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive.h、mi_ive.h、mi_ive.h
- 库文件：libive.a

- 注意

计算公式如下：

$$O(h) = \begin{cases} 1, h \geq u16HorTimes \\ 0, h < u16HorTimes \end{cases}$$

$$O(v) = \begin{cases} 1, v \geq u16VerTimes \\ 0, v < u16VerTimes \end{cases}$$

h/v 对应到水平/垂直次数，O(h)/ O(v) 对应到水平/垂直输入图像。

- 举例

无。

1.41. MI_IVE_Acc

- 功能

执行两灰度图像的累积运算任务。

- 语法

MI_S32 MI_IVE_Acc (MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t * pstSrc0, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_DstImage_t *pstDst, MVE_IVE_AccCtrl_t * pstAccCtrl, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc0	源图像 1 指针。不能为空。	输入
pstSrc1	源图像 2 指针。不能为空。	输入
pstDst	输t出图像指针。不能为空。	输出
pstAccCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc0	U8C1	16 byte	64x64～1920x1080
pstSrc1	U8C1	16 byte	64x64～1920x1080
pstDst	U8C1	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

➢ 注意

计算公式如下 ：

E_MI_IVE_ACC_MODE_INCREASE：

$$I_{out}(x,y) = I_{src0}(x,y) + I_{src1}(x,y)$$

E_MI_IVE_ACC_MODE_DECREASE：

$$I_{out}(x,y) = I_{src0}(x,y) - I_{src1}(x,y)$$

E_MI_IVE_ACC_MODE_INCREASE_MAP_255TO1：

$$I_{out}(x,y) = I_{src0}(x,y) + (I_{src1}(x,y)\ \&\ 0x1)$$

➢ 举例

无。

1.42. MI_IVE_Matrix_Transform

➢ 功能

执行矩阵运算任务。

➢ 语法

MI_S32 MI_IVE_Matrix_Transform(MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_SrcImage_t *pstSrc3, MI_IVE_DstImage_t *pstDst1, MI_IVE_DstImage_t *pstDst2, MI_IVE_DstImage_t *pstDst3, MI_IVE_MatrTranfCtrl_t *pstMatrTranfCtrl, MI_BOOL bInstant);

➢ 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。	输入

参数名称	描述	输入/输出
	取值范围：[0, MI_IVE_HANDLE_MAX)。	
pstSrc1	源图像指针。不能为空。	输入
pstSrc2	源图像指针。不能为空。	输入
pstSrc3	源图像指针。不能为空。	输入
pstDst1	输t出图像指针。不能为空。	输出
pstDst2	输t出图像指针。不能为空。	输出
pstDst3	输t出图像指针。不能为空。	输出
pstMatrTranfCtrl	控制信息指针。不能为空。	输入
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	S32C1	16 byte	64x64～1920x1080
pstSrc2	S32C1	16 byte	64x64～1920x1080
pstSrc3	S32C1	16 byte	64x64～1920x1080
pstDst1	S32C1	16 byte	64x64～1920x1080
pstDst2	S32C1	16 byte	64x64～1920x1080
pstDst3	S32C1	16 byte	64x64～1920x1080

- 返回值

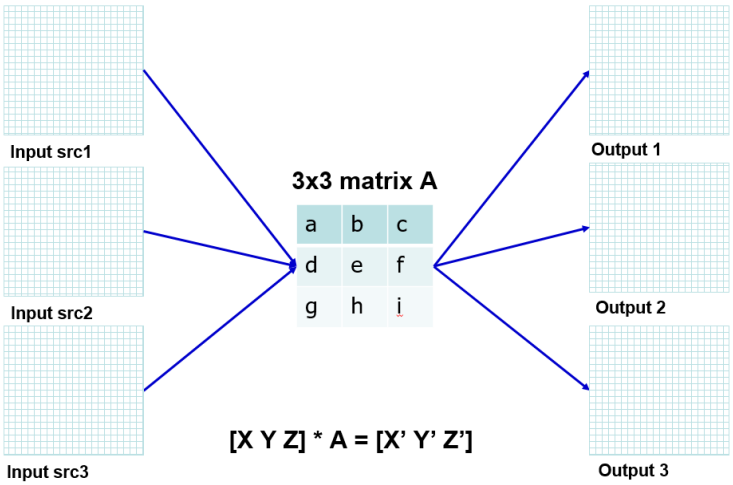
返回值	描述
0	成功。
非 0	失败，参见错误码。

- 依赖

  - 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
  - 库文件：libive. a

- 注意

  下方为计算图解示意图



每一个输入源的元素都是由16个整数,15个小数和1个符号所组成的。
每一个矩阵的元素都是由1个整数,14个小数和1个符号所组成的。

- 举例

无。

1.43. MI_IVE_Image_Dot

- 功能

执行图像点乘积任务。

- 语法

MI_S32 MI_IVE_ Image_Dot (MI_IVE_HANDLE hHandle, MI_IVE_SrcImage_t *pstSrc1, MI_IVE_SrcImage_t *pstSrc2, MI_IVE_DstImage_t *pstDst, MI_BOOL bInstant);

- 形参

参数名称	描述	输入/输出
hHandle	区域句柄号。 取值范围：[0，MI_IVE_HANDLE_MAX)。	输入
pstSrc1	源图像指针。不能为空。	输入
pstSrc2	源图像指针。不能为空。	输入
pstDst	输t出图像指针。不能为空。	输出
bInstant	保留。	输入

参数名称	支持图像类型	地址对齐	分辨率
pstSrc1	S32C1	16 byte	64x64～1920x1080
pstSrc2	S32C1	16 byte	64x64～1920x1080
pstDst	S32C1	16 byte	64x64～1920x1080

➢ 返回值

返回值	描述
0	成功。
非 0	失败，参见错误码。

➢ 依赖

- 头文件：mi_comm_ive. h、mi_ive. h、mi_ive. h
- 库文件：libive. a

➢ 注意

计算公式如下

$$I_{out} = I_{src1}(x,y) \times I_{src2}(x,y)$$

每一个输入源的元素都是由16个整数,15个小数和1个符号所组成的。

➢ 举例

无。

2. IVE 数据类型

IVE 相关数据类型、数据结构定义如下：

MI_IVE_HIST_NUM	定义直方图统计 bin 数目
MI_IVE_MAP_NUM	定义映射查找表项数目
MI_IVE_MAX_REGION_NUM	定义最大连通区域数目
MI_IVE_ST_MAX_CORNER_NUM	定义 Shi-Tomasi-like 角点最大数目
MI_IVE_ImageType_e	定义二维广义图像支持的图像类型
MI_IVE_Image_t	定义二维广义图像信息。
MI_IVE_SrcImage_t	定义源图像
MI_IVE_DstImage_t	定义输出图像
MI_IVE_Data_t	定义以 byte 为单位的二维图像信息。
MI_IVE_SrcData_t	定义以 byte 为单位的二维源数据信息
MI_IVE_DstData_t	定义 byte 为单位的二维输出数据信息
MI_IVE_MemInfo_t	定义一维数据内存信息
MI_IVE_SrcMemInfo_t	定义一维源数据
MI_IVE_DstMemInfo_t	定义一维输出数据
MI_IVE_Length8bit_u	定义 8bit 数据共享体
MI_IVE_PointU16_t	定义 U16 表示的点信息结构体
MI_IVE_PointS25Q7_t	定义 S25Q7 定点表示的点信息结构体
MI_IVE_Rect_t	定义 U16 表示的矩形信息结构体
MI_IVE_FilterCtrl_t	定义模板滤波控制信息
MI_IVE_CscMode_e	定义色彩空间转换模式
MI_IVE_CscCtrl_t	定义色彩空间转换控制信息
MI_IVE_FilterAndCscCtrl_t	定义模板滤波加色彩空间转换复合功能控制信息
MI_IVE_SobelOutCtrl_e	定义 sobel 输出控制信息
MI_IVE_SobelCtrl_t	定义 sobel 边缘提取控制信息
MI_IVE_MagAndAngOutCtrl_e	定义 canny 边缘幅值与角度计算的输出格式
MI_IVE_MagAndAngCtrl_t	定义 canny 边缘幅值和幅角计算的控制信息
MI_IVE_DilateCtrl_t：	定义膨胀控制信息
MI_IVE_ErodeCtrl_t	定义腐蚀控制信息
MI_IVE_ThreshMode_e	定义图像二值化输出格式
MI_IVE_ThreshCtrl_t	定义图像二值化控制信息
MI_IVE_SubMode_e	定义两图像相减输出格式
MI_IVE_SubCtrl_t	定义两图像相减控制参数

MI_IVE_IntegOutCtrl_e	定义积分图输出控制参数
MI_IVE_IntegCtrl_t	定义积分图计算控制参数
MI_IVE_ThreshS16Mode_e	定义 16bit 有符号图像的阈值化模式
MI_IVE_ThreshS16Ctrl_t	定义 16bit 有符号图像的阈值化控制参数
MI_IVE_ThreshU16Mode_e	定义 16bti 无符号图像的阈值化模式
MI_IVE_ThreshU16Ctrl_t	定义 16bit 无符号图像的阈值化控制参数
MI_IVE_16BitTo8BitMode_e	定义 16bit 图像到 8bit 图像的转化模式
MI_IVE_16bitTo8BitCtrl_t	定义 16bit 图像到 8bit 图像的转化控制参数
MI_IVE_OrdStatFilterMode_e	定义顺序统计量滤波模式
MI_IVE_OrdStatFilterCtrl_t	定义顺序统计量滤波控制参数
MI_IVE_MapLutMem_t	定义 Map 算子的查找表内存信息
MI_IVE_EqualizeHistCtrlMem_t	定义直方图衡化辅助均内存。
MI_IVE_EqualizeHistCtrl_t	定义直方图均衡化控制参数。
MI_IVE_AddMode_e	定义 add 计算模式
MI_IVE_AddCtrl_t	定义两图像的加权加控制参数
MI_IVE_NccDstMem_t	定义 NCC 的输出内存信息。
MI_IVE_Region_t	定义连通区域信息。
MI_IVE_CcBlob_t	定义连通区域标记的输出信息。
MI_IVE_CclMode_e	定义连通区域模式。
MI_IVE_CclCtrl_t:	定义连通区域标记控制参数。
MI_IVE_GmmCtrl_t	定义 GMM 背景建模的控制参数
MI_IVE_CannyStackSize_t	定义 Canny 边缘前半部分计算时强边缘点栈大小结构体
MI_IVE_CannyHysEdgeCtrl_t_	定义 Canny 边缘前半部分计算任务的控制参数。
MI_IVE_LbpCmpMode_e	定义 LBP 计算的比较模式。
MI_IVE_LbpChalMode_e	定义 LBP 输入信道模式。
MI_IVE_LbpCtrrl_t	定义 LBP 纹理计算控制参数
MI_IVE_NormGradOutCtrl_e	定义归一化梯度信息计算任务输出控制枚举类型
IVE_NormGradCtrl_t	定义归一化梯度信息计算控制参数
MI_IVE_MvS9Q7_t	定义位移结构体
MI_IVE_LkOpticalFlowCtrl_t	定义 LK 光流计算控制参数
MI_IVE_SadMode_e	定义 SAD 计算模式。
MI_IVE_SadOutCtrl_e	定义 SAD 输出控制模式。
MI_IVE_SadCtrl_t	定义 SAD 控制参数。
MVE_IVE_BernsenCtrl_t	定义 Bernsen 控制参数。
MVE_IVE_BernsenMode_e	定义 Bernsen 阀值模式。
MVE_IVE_LineFilterHorCtrl_t	定义 LineFilterHor 控制参数。
MVE_IVE_LineFilterVerCtrl_t	定义 LineFilterVer 控制参数。
MVE_IVE_NoiseRemoveHorCtrl_t	定义 NoiseRemoveHor 控制参数。
MVE_IVE_NoiseRemoveVerCtrl_t	定义 NoiseRemoveVer 控制参数。
MVE_IVE_AdpThreshCtrl_t	定义 AdpThresh 控制参数。

MVE_IVE_ResizeCtrl_t	定义 Resize 控制参数。
MVE_IVE_ResizeMode_e	定义 Resize 输入模式。
MVE_IVE_BatCtrl_t	定义 SAD 控制参数。
MVE_IVE_BatMode_e	定义 Bat 运算模式。
MVE_IVE_AccCtrl_t	定义 Acc 控制参数。
MVE_IVE_AccMode_e	定义 Acc 运算模式。
MI_IVE_MatrTranfMode_e	定义 matrix_transform 的输入信道模式。
MI_IVE_MatrTranfCtrl_t	定义 matrix_transform 控制参数。

2.1. 定点数据类型

- 说明

  定义定点化的数据类型。

- 定义

```
typedef  unsigned  char   MI_U0Q8;
typedef  unsigned  char   MI_U1Q7;
typedef  unsigned  char   MI_U5Q3;
typedef  unsigned  short  MI_U0Q16;
typedef  unsigned  short  MI_U4Q12;
typedef  unsigned  short  MI_U6Q10;
typedef  unsigned  short  MI_U8Q8;
typedef  unsigned  short  MI_U14Q2;
typedef  unsigned  short  MI_U12Q4;
typedef  short            MI_S14Q2;
typedef  short            MI_S9Q7;
typedef  unsigned  int    MI_U22Q10;
typedef  unsigned  int    MI_U25Q7;
typedef  int              MI_S25Q7;
typedef  unsigned  short  MI_U8Q4F4;  /*8bits unsigned integer，4bits decimal fraction，
                                        4bits  flag  bits*/
```

- 成员

成员名称	描述
MI_U0Q8	用0bit 表示整数部分，8bit 表示小数部分。文档中用 UQ0.8 来表示。
MI_U1Q7	用高1bit 无符号数据表示整数部分，低 7bit 表示小数部分。文档中用UQ1.7 来表示。
MI_U5Q3	用高5bit 无符号数据表示整数部分，低 3bit 表示小数部分。文档中用UQ5.3 来表示。
MI_U0Q16	用0bit 表示整数部分，16bit 表示小数部分。文档中用 UQ0.16 来表示。
MI_U4Q12	用高4bit 无符号数据表示整数部分，低 12bit 表示小数部分。文档中用 UQ4.12 来表示。
MI_U6Q10	用高6bit 无符号数据表示整数部分，低 10bit 表示小数部分。文档中用 UQ6.10 来表示。
MI_U8Q8	用高8bit 无符号数据表示整数部分，低 8bit 表示小数部分。文档中用UQ8.8 来表示。
MI_U14Q2	用高14bit 无符号数据表示整数部分，低 2bit 表示小数部分。文档中用 UQ14.2 来表示。
MI_U12Q4	用高12bit 无符号数据表示整数部分，低 4bit 表示小数部分。文档中用 UQ12.4 来表示。
MI_S14Q2	用高14bit 有符号数据表示整数部分，低 2bit 表示小数部分。文档中用 SQ14.2 来表示。

成员名称	描述
MI_S9Q7	用高 9bit 有符号数据表示整数部分，低 7bit 表示小数部分。文档中用SQ9.7 来表示。
MI_U22Q10	用高 22bit 无符号数据表示整数部分，低 10bit 表示小数部分。文档中用 UQ22.10 来表示。
MI_U25Q7	用高 25bit 无符号数据表示整数部分，低 7bit 表示小数部分。文档中用 UQ25.7 来表示。
MI_S25Q7	用高 25bit 有符号数据表示整数部分，低 7bit 表示小数部分。文档中用 SQ25.7 来表示。
MI_U8Q4F4	用高 8bit 无符号数据表示整数部分，中间 4bit 表示小数部分，低 4bit 表示标志位。文档中用 UQF8.4.4 来表示。

※ 注意事项

MI_UxQyFz\MI_SxQy：

- U 后面的数字 x 表示是用 x bit 无符号数据表示整数部分；
- S 后面的数字 x 表示用 x bit 有符号数据表示整数部分；
- Q 后面的数字 y 表示用 y bit 数据表示小数部分；
- F 后面的数字 z表示用 z bit 来表示标志位；
- 从左到右依次表示高 bit 位到低 bit 位。

➢ 相关数据类型及接口

无。

2.2. MI_IVE_HIST_NUM

➢ 说明

定义直方图统计 bin 数目。

➢ 定义

#define MI_IVE_HIST_NUM 256

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.3. MI_IVE_MAP_NUM

- 说明

  定义映射查找表项数目。

- 定义

  #define MI_IVE_MAP_NUM256

- 成员

  无。

※ 注意事项

  无。

- 相关数据类型及接口

  无。

2.4. MI_IVE_MAX_REGION_NUM

- 说明

  定义最大连通区域数目。

- 定义

  #define MI_IVE_MAX_REGION_NUM 255

- 成员

  无。

※ 注意事项

  无。

- 相关数据类型及接口

  无。

2.5. MI_IVE_ST_MAX_CORNER_NUM

- 说明

  定义 Shi-Tomasi-like 角点最大数目。

- 定义

```
#define MI_IVE_ST_MAX_CORNER_NUM   200
```

- 成员

  无。

※ 注意事项

  无。

- 相关数据类型及接口

  无。

2.6. MI_IVE_MASK_SIZE_5X5

- 说明

  定义 Mask size。

- 定义

```
#define MI_IVE_MASK_SIZE_5X5      25
```

- 成员

  无。

※ 注意事项

  无。

- 相关数据类型及接口

  无。

2.7. MI_IVE_CANNY_STACK_RESERVED_SIZE

- 说明

  定义 Canny Stack Reserved Size。

- 定义

```
#define MI_IVE_CANNY_STACK_RESERVED_SIZE    12
```

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.8. MI_IVE_ImageType_e

➢ 说明

定义二维广义图像支持的图像类型。

➢ 定义

```
typedef enum
{
    E_MI_IVE_IMAGE_TYPE_U8C1            =0x0,
    E_MI_IVE_IMAGE_TYPE_S8C1            =0x1,
    E_MI_IVE_IMAGE_TYPE_YUV420SP        =0x2, /*YUV420 SemiPlanar*/
    E_MI_IVE_IMAGE_TYPE_YUV422SP        =0x3, /*YUV422 SemiPlanar*/
    E_MI_IVE_IMAGE_TYPE_YUV420P         =0x4, /*YUV420 Planar*/
    E_MI_IVE_IMAGE_TYPE_YUV422P         =0x5, /*YUV422 planar*/
    E_MI_IVE_IMAGE_TYPE_S8C2_PACKAGE    =0x6,
    E_MI_IVE_IMAGE_TYPE_S8C2_PLANAR     =0x7,
    E_MI_IVE_IMAGE_TYPE_S16C1           =0x8,
    E_MI_IVE_IMAGE_TYPE_U16C1           =0x9,
    E_MI_IVE_IMAGE_TYPE_U8C3_PACKAGE    =0xa,
    E_MI_IVE_IMAGE_TYPE_U8C3_PLANAR     =0xb,
    E_MI_IVE_IMAGE_TYPE_S32C1           =0xc,
    E_MI_IVE_IMAGE_TYPE_U32C1           =0xd,
    E_MI_IVE_IMAGE_TYPE_S64C1           =0xe,
    E_MI_IVE_IMAGE_TYPE_U64C1           =0xf,
    E_MI_IVE_IMAGE_TYPE_BUTT
}MI_IVE_ImageType_e;
```

➢ 成员

成员名称	描述
E_MI_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。
E_MI_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。

成员名称	描述
E_MI_IVE_IMAGE_TYPE_YUV420SP	YUV420 Semiplanar 格式的图像。请参见图 1-3。
E_MI_IVE_IMAGE_TYPE_YUV422SP	YUV422 Semiplanar 格式的图像。请参见图 1-4。
E_MI_IVE_IMAGE_TYPE_YUV420P	YUV420 Planar 格式的图像。请参见图 1-5。
E_MI_IVE_IMAGE_TYPE_YUV422P	YUV422 Planar 格式的图像。请参见图 1-6。
E_MI_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。
E_MI_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。
E_MI_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。
E_MI_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。
E_MI_IVE_IMAGE_TYPE_U8C3_PACKAGE	每个像素用 3 个 8bit 无符号数据表示且以 planar 格式存储 3 通道图像。请参见图 1-9。
E_MI_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。
E_MI_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。
E_MI_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。
E_MI_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。
E_MI_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。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_Image_t
- MI_IVE_SrcImage_t
- MI_IVE_DstImage_t

2.9. MI_IVE_Image_t

➢ 说明

定义二维广义图像信息。

➢ 定义

```
typedef struct MI_IVE_Image_s
{
    MI_IVE_ImageType_e eType;
    MI_PHY aphyPhyAddr[3];
    MI_U8 *apu8VirAddr[3];
    MI_U16 au16Stride[3];
    MI_U16 u16Width;
    MI_U16 u16Height;
    MI_U16 u16Reserved;        /*Can be used such as elemSize*/
}MI_IVE_Image_t;
```

➢ 成员

成员名称	描述
enType	广义图像的图像类型。
aphyPhyAddr[3]	广义图像的物理地址数组。
apu8VirAddr[3]	广义图像的虚拟地址数组。
au16Stride[3]	广义图像的跨度。
u16Width	广义图像的宽度。
u16Height	广义图像的高度。
u16Reserved	保留位。

※ 注意事项

- 不同的算子对图像图像的输入输出地址是否对齐有不同的要求。
- u16Width、u16Height 和 u16Stride 均是以像素为度量单位的。
- 每种type 下的图像示意图请参见图1-2～图1-10。

➢ 相关数据类型及接口

- MI_IVE_ImageType_e
- MI_IVE_SrcImage_t
- MI_IVE_DstImage_t

2.10. MI_IVE_SrcImage_t

➢ 说明

定义源图像。

➢ 定义

typedef MI_IVE_Image_t MI_IVE_SrcImage_t;

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_Image_t
- MI_IVE_DstImage_t

2.11. MI_IVE_DstImage_t

➢ 说明

定义输出图像。

➢ 定义

typedef MI_IVE_Image_t MI_IVE_DstImage_t;

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_Image_t

MI_IVE_SrcImage_t

2.12. MI_IVE_Data_t

➢ 说明

定义以 byte 为单位的二维数据信息。

➢ 定义

```
typedef struct MI_IVE_Data_s
{
    MI_PHY phyPhyAddr; /*Physical address of the data*/
    MI_U8  *pu8VirAddr;
    MI_U16 u16Stride;      /*Data stride by byte*/
    MI_U16 u16Height;      /*Data height by byte*/
    MI_U16 u16Width;       /*Data width by byte*/
    MI_U16 u16Reserved;
}MI_IVE_Data_t;
```

➢ 成员

成员名称	描述
phyPhyAddr	图像物理地址。
pu8VirAddr	图像虚拟地址。
u16Stride	图像跨度。
u16Height	图像宽度。
u16Width	图像高度。
u16Reserved	保留位。

※ 注意事项

表示以 byte 为单位的二维数据；可以与 MI_IVE_Image_t 图像进行转换。

➢ 相关数据类型及接口

无。

2.13. MI_IVE_SrcData_t

➢ 说明

定义以 byte 为单位的二维源数据信息。

➢ 定义

```
typedef MI_IVE_Data_t MI_IVE_SrcData_t;
```

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_Image_t
- MI_IVE_DstData_t

2.14. MI_IVE_DstData_t

➢ 说明

定义 byte 为单位的二维输出数据信息。

➢ 定义

typedef MI_IVE_Data_t MI_IVE_DstData_t;

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_Image_t
- MI_IVE_SrcImage_t

2.15. MI_IVE_MemInfo_t

➢ 说明

定义一维数据内存信息。

➢ 定义

```
typedef struct MI_IVE_MemInfo_s
{
    MI_PHY phyPhyAddr;
    MI_U8 *pu8VirAddr;
    MI_U32 u32Size;
}MI_IVE_MemInfo_t;
```

➢ 成员

成员名称	描述
phyPhyAddr	一维数据物理地址。
pu8VirAddr	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

- MI_IVE_SrcMemInfo_t
- MI_IVE_DstMemInfo_t

2.16. MI_IVE_SrcMemInfo_t

➢ 说明

定义一维源数据。

➢ 定义

typedef MI_IVE_MemInfo_t MI_IVE_SrcMemInfo_t;

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_MemInfo_t
- MI_IVE_DstMemInfo_t

2.17. MI_IVE_DstMemInfo_t

➢ 说明

定义一维输出数据。

➢ 定义

typedef MI_IVE_MemInfo_t MI_IVE_DstMemInfo_t;

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

- MI_IVE_MemInfo_t
- MI_IVE_SrcMemInfo_t

2.18. MI_IVE_Length8bit_u

➢ 说明

定义 8bit 数据联合体。

➢ 定义

```
typedef union
{
    MI_S8 s8Val;
    MI_U8 u8Val;
} MI_IVE_Length8bit_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。

2.19. MI_IVE_PointU16_t

➢ 说明

定义 U16 表示的点信息结构体。

➢ 定义

```
typedef struct MI_IVE_PointU16_s
{
    MI_U16 u16X;
    MI_U16 u16Y;
}MI_IVE_PointU16_t;
```

➢ 成员

成员名称	描述
u16X	点的 x 坐标。
u16Y	点的 y 坐标。

※ 注意事项

无。

➢ 相关数据类型及接口

无

2.20. MI_IVE_PointS25Q7_t

➢ 说明

定义 S25Q7 定点表示的点信息结构体。

➢ 定义

```
typedef struct MI_IVE_PointS25Q7_s
{
    MI_S25Q7    s25q7X;          /*X coordinate*/
    MI_S25Q7    s25q7Y;          /*Y coordinate*/
}MI_IVE_PointS25Q7_t;
```

➢ 成员

成员名称	描述
s25q7X	点的 x 坐标，以 SQ25.7 表示。
s25q7Y	点的 y 坐标，以 SQ25.7 表示。

※ 注意事项

无。

➢ 相关数据类型及接口

无

2.21. MI_IVE_Rect_t

➢ 说明

定义 U16 表示的矩形信息结构体。

➢ 定义

```
typedef struct MI_IVE_Rect_s
{
    MI_U16 u16X;
    MI_U16 u16Y;
    MI_U16 u16Width;
    MI_U16 u16Height;
}MI_IVE_Rect_t;
```

➢ 成员

成员名称	描述
u16X	矩形相对于坐标原点最近点的 x 坐标。
u16Y	矩形相对于坐标原点最近点的 y 坐标。
u16Width	矩形的宽。
u16Height	矩形的高。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.22. MI_IVE_FilterCtrl_t

➢ 说明

定义模板滤波控制信息。

➢ 定义

```
typedef struct MI_IVE_FilterCtrl_s
{
    MI_S8 as8Mask[MI_IVE_MASK_SIZE_5X5];       /*Template parameter filter coefficient*/
    MI_U8 u8Norm;                /*Normalization parameter, by right shift*/
}MI_IVE_FilterCtrl_t;
```

➢ 成员

成员名称	描述
as8Mask[MI_IVE_MASK_SIZE_5X5]	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。

2.23. MI_IVE_CscMode_e

➢ 说明

定义色彩空间转换模式。

➢ 定义

```
typedef enum
{
/*CSC: YUV2RGB, video transfer mode, RGB value range [16, 235]*/
E_MI_IVE_CSC_MODE_VIDEO_BT601_YUV2RGB     = 0x0,
/*CSC: YUV2RGB, video transfer mode, RGB value range [16, 235]*/
E_MI_IVE_CSC_MODE_VIDEO_BT709_YUV2RGB     = 0x1,
/*CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255]*/
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2RGB   = 0x2,
/*CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255]*/
```

```
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2RGB        = 0x3,
/*CSC: YUV2HSV, picture transfer mode, HSV value range [0, 255]*/
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2HSV  = 0x4,
/*CSC: YUV2HSV, picture transfer mode, HSV value range [0, 255]*/
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2HSV  = 0x5,
/*CSC: YUV2LAB, picture transfer mode, Lab value range [0, 255]*/
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2LAB  = 0x6,
/*CSC: YUV2LAB, picture transfer mode, Lab value range [0, 255]*/
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2LAB  = 0x7,
/*CSC: RGB2YUV, video transfer mode, YUV value range [0, 255]*/
E_MI_IVE_CSC_MODE_VIDEO_BT601_RGB2YUV    = 0x8,
/*CSC: RGB2YUV, video transfer mode, YUV value range [0, 255]*/
E_MI_IVE_CSC_MODE_VIDEO_BT709_RGB2YUV    = 0x9,
/*CSC: RGB2YUV, picture transfer mode, Y:[16, 235], U\V:[16, 240]*/
E_MI_IVE_CSC_MODE_PIC_BT601_RGB2YUV  = 0xa,
/*CSC: RGB2YUV, picture transfer mode, Y:[16, 235], U\V:[16, 240]*/
E_MI_IVE_CSC_MODE_PIC_BT709_RGB2YUV  = 0xb,
E_MI_IVE_CSC_MODE_BUTT
}MI_IVE_CscMode_e;
```

- 成员

成员名称	描述
E_MI_IVE_CSC_MODE_VIDEO_BT601_YUV2RGB	BT601 的 YUV2RGB 视频变换。
E_MI_IVE_CSC_MODE_VIDEO_BT709_YUV2RGB	BT709 的 YUV2RGB 视频变换。
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2RGB	BT601 的 YUV2RGB 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2RGB	BT709 的 YUV2RGB 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2HSV	BT601 的 YUV2HSV 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2HSV	BT709 的 YUV2HSV 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT601_YUV2LAB	BT601 的 YUV2LAB 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT709_YUV2LAB	BT709 的 YUV2LAB 图像变换。
E_MI_IVE_CSC_MODE_VIDEO_BT601_RGB2YUV	BT601 的 RGB2YUV 视频变换。
E_MI_IVE_CSC_MODE_VIDEO_BT709_RGB2YUV	BT709 的 RGB2YUV 视频变换。
E_MI_IVE_CSC_MODE_PIC_BT601_RGB2YUV	BT601 的 RGB2YUV 图像变换。
E_MI_IVE_CSC_MODE_PIC_BT709_RGB2YUV	BT709 的 RGB2YUV 图像变换。

※ 注意事项

- E_MI_IVE_CSC_MODE_VIDEO_BT601_YUV2RGB 和 E_MI_IVE_CSC_MODE_VIDEO_BT709_YUV2RGB 模式，输出满足 16≤R、G、B≤235。
- E_MI_IVE_CSC_MODE_PIC_BT601_YUV2RGB 和 E_MI_IVE_CSC_MODE_PIC_BT709_YUV2RGB 模式，输出满足 0≤R、G、B≤255。
- E_MI_IVE_CSC_MODE_PIC_BT601_YUV2HSV 和 E_MI_IVE_CSC_MODE_PIC_BT709_YUV2HSV 模式，输出满足 0≤H、S、V≤255。
- E_MI_IVE_CSC_MODE_PIC_BT601_YUV2LAB 和 E_MI_IVE_CSC_MODE_PIC_BT709_YUV2LAB 模式，输出满足 0≤L、A、B≤255。
- E_MI_IVE_CSC_MODE_VIDEO_BT601_RGB2YUV 和 E_MI_IVE_CSC_MODE_VIDEO_BT709_RGB2YUV 模式，输出满足 0≤Y、U、V≤255。
- E_MI_IVE_CSC_MODE_PIC_BT601_RGB2YUV 和 E_MI_IVE_CSC_MODE_PIC_BT709_RGB2YUV 模式，输出满足 0≤Y≤235，0≤U、V≤240。

➢ 相关数据类型及接口

- MI_IVE_CscCtrl_t
- MI_IVE_FilterAndCscCtrl_t

2.24. MI_IVE_CscCtrl_t

➢ 说明

定义色彩空间转换控制信息。

➢ 定义

```
typedef struct MI_IVE_CscCtrl_s
{
    MI_IVE_CscMode_e enMode; /*Working mode*/
}MI_IVE_CscCtrl_t;
```

➢ 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

MI_IVE_CscMode_e

2.25. MI_IVE_FilterAndCscCtrl_t

➢ 说明

定义模板滤波加色彩空间转换复合功能控制信息。

- 定义

```
typedef struct MI_IVE_FilterAndCscCtrl_s
{
    MI_IVE_CscMode_e    eMode; /*CSC working mode*/
    MI_S8    as8Mask[MI_IVE_MASK_SIZE_5X5];    /*Template parameter filter coefficient*/
    MI_U8    u8Norm;            /*Normalization parameter, by right shift*/
}MI_IVE_FilterAndCscCtrl_t;
```

- 成员

成员名称	描述
eMode	工作模式。
as8Mask[MI_IVE_MASK_SIZE_5X5]	5x5 模板系数。
u8Norm	归一化参数。 取值范围：[0, 13]。

※ 注意事项

仅支持 YUV2RGB 的 4 种模式。

- 相关数据类型及接口

MI_IVE_CscMode_e

2.26. MI_IVE_SobelOutCtrl_e

- 说明

定义 Sobel 输出控制信息。

- 定义

```
typedef enum
{
    E_MI_IVE_SOBEL_OUT_CTRL_BOTH = 0x0, /*Output horizontal and vertical*/
    E_MI_IVE_SOBEL_OUT_CTRL_HOR = 0x1, /*Output horizontal*/
    E_MI_IVE_SOBEL_OUT_CTRL_VER = 0x2, /*Output vertical*/
    E_MI_IVE_SOBEL_OUT_CTRL_BUTT
}MI_IVE_SobelOutCtrl_e;
```

- 成员

成员名称	描述
E_MI_IVE_SOBEL_OUT_CTRL_BOTH	同时输出用模板和转置模板滤波的结果。
E_MI_IVE_SOBEL_OUT_CTRL_HOR	仅输出用模板直接滤波的结果。
E_MI_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

MI_IVE_SobelCtrl_t

2.27. MI_IVE_SobelCtrl_t

➢ 说明

定义 Sobel-like 梯度计算控制信息。

➢ 定义

```
typedef struct MI_IVE_SobelCtrl_s
{
    MI_IVE_SobelOutCtrl_e eOutCtrl; /*Output format*/
    MI_S8 as8Mask[MI_IVE_MASK_SIZE_5X5]; /*Template parameter*/
}MI_IVE_SobelCtrl_t;
```

➢ 成员

成员名称	描述
eOutCtrl	输出控制枚举参数。
as8Mask[MI_IVE_MASK_SIZE_5X5]	5x5 模板系数。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_SobelOutCtrl_e

2.28. MI_IVE_MagAndAngOutCtrl_e

➢ 说明

定义梯度幅值与角度计算的输出格式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_MAG_AND_ANG_OUT_CTRL_MAG
     =                                          0x0,
    E_MI_IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_
    ANG  =                                      0x1,
    E_MI_IVE_MAG_AND_ANG_OUT_CTRL_BUTT
}MI_IVE_MagAndAngOutCtrl_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_MAG_AND_ANG_OUT_CTRL_MAG	仅输出幅值。
E_MI_IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG	同时输出幅值和角度值。

※ 注意事项

无。

➢ 相关数据类型及接口

IVE_MAG_AND_ANG_CTRL_S

2.29. MI_IVE_MagAndAngCtrl_t

➢ 说明

定义梯度幅值和幅角计算的控制信息。

➢ 定义

```
typedef struct MI_IVE_MagAndAngCtrl_s
{
    MI_IVE_MagAndAngOutCtrl_e eOutCtrl;
    MI_U16 u16Thr;
    MI_S8 as8Mask[MI_IVE_MASK_SIZE_5X5]; /*Template parameter.*/
}MI_IVE_MagAndAngCtrl_t;
```

➢ 成员

成员名称	描述
eOutCtrl	输出格式。
u16Thr	用于对幅值进行阈值化的阈值。
as8Mask[MI_IVE_MASK_SIZE_5X5]	5x5 模板系数。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_MagAndAngOutCtrl_e

2.30. MI_IVE_DilateCtrl_t

➢ 说明

定义膨胀控制信息。

➢ 定义

```
typedef struct MI_IVE_DilateCtrl_s
{
    MI_U8 au8Mask[MI_IVE_MASK_SIZE_5X5]; /*The template parameter value must be 0 or 255.*/
}MI_IVE_DilateCtrl_t;
```

➢ 成员

成员名称	描述
au8Mask[MI_IVE_MASK_SIZE_5X5]	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。

2.31. MI_IVE_ErodeCtrl_t

➢ 说明

定义腐蚀控制信息。

➢ 定义

```
typedef struct MI_IVE_ErodeCtrl_s
{
    MI_U8 au8Mask[MI_IVE_MASK_SIZE_5X5]; /*The template parameter value must be 0 or 255.*/
}MI_IVE_ErodeCtrl_t;
```

➢ 成员

成员名称	描述
au8Mask[MI_IVE_MASK_SIZE_5X5]	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。

2.32. MI_IVE_ThreshMode_e

➢ 说明

定义图像二值化输出格式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_THRESH_MODE_BINARY = 0x0,      /*srcVal <= lowThr, dstVal = minVal; srcVal > lowThr, dstVal = maxVal.*/
    E_MI_IVE_THRESH_MODE_TRUNC = 0x1,       /*srcVal <= lowThr, dstVal = srcVal; srcVal > lowThr, dstVal = maxVal.*/
    E_MI_IVE_THRESH_MODE_TO_MINVAL = 0x2,   /*srcVal <= lowThr, dstVal = minVal; srcVal > lowThr, dstVal = srcVal.*/
    E_MI_IVE_THRESH_MODE_MIN_MID_MAX = 0x3, /*srcVal <= lowThr, dstVal = minVal; lowThr < srcVal <= highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal.*/
    E_MI_IVE_THRESH_MODE_ORI_MID_MAX = 0x4, /*srcVal <= lowThr, dstVal = srcVal; lowThr < srcVal <= highThr, dstVal =
```

```
                                                midVal; srcVal > highThr, dstVal =
                                                maxVal. */
    E_MI_IVE_THRESH_MODE_MIN_MID_ORI            /*srcVal <= lowThr, dstVal = minVal;
    = 0x5,                                      lowThr < srcVal <= highThr, dstVal =
                                                midVal; srcVal > highThr, dstVal =
                                                srcVal. */
    E_MI_IVE_THRESH_MODE_MIN_ORI_MAX            /*srcVal <= lowThr, dstVal = minVal;
    = 0x6,                                      lowThr < srcVal <= highThr, dstVal =
                                                srcVal; srcVal > highThr, dstVal =
                                                maxVal. */
    E_MI_IVE_THRESH_MODE_ORI_MID_ORI =          /*srcVal <= lowThr, dstVal = srcVal;
    0x7,                                        lowThr < srcVal <= highThr, dstVal =
                                                midVal; srcVal > highThr, dstVal =
                                                srcVal. */
    E_MI_IVE_THRESH_MODE_BUTT
}MI_IVE_ThreshMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_THRESH_MODE_BINARY	srcVal ≤ lowThr, dstVal = minVal; srcVal > lowThr, dstVal = maxVal。
E_MI_IVE_THRESH_MODE_TRUNC	srcVal ≤ lowThr, dstVal = srcVal; srcVal > lowThr, dstVal = maxVal。
E_MI_IVE_THRESH_MODE_TO_MINVAL	srcVal ≤lowThr, dstVal = minVal; srcVal > lowThr, dstVal = srcVal。
E_MI_IVE_THRESH_MODE_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal。
E_MI_IVE_THRESH_MODE_ORI_MID_MAX	srcVal ≤lowThr, dstVal = srcVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal。
E_MI_IVE_THRESH_MODE_MIN_MID_ORI	srcVal ≤lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = midVal; srcVal > highThr, dstVal = srcVal。
E_MI_IVE_THRESH_MODE_MIN_ORI_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤ highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal。
E_MI_IVE_THRESH_MODE_ORI_MID_ORI	srcVal≤ lowThr, dstVal = srcVal; lowThr < srcVal ≤ highThr, dstVal = midVal; srcVal > highThr, dstVal = srcVal。

※ 注意事项

计算公式请参见 MI_IVE_Thresh 中的【注意】，示意图请参见图 2-8。

➢ 相关数据类型及接口

MI_IVE_ThreshCtrl_t

2.33. MI_IVE_ThreshCtrl_t

➢ 说明

定义图像二值化控制信息。

➢ 定义

```
typedef struct MI_IVE_ThreshCtrl_s
{
    MI_IVE_ThreshMode_e eMode;
    MI_U8 u8LowThr; /*user-defined threshold, 0<=u8LowThr<=255 */
    MI_U8 u8HighThr; /*user-defined threshold, if
    eMode<E_MI_IVE_THRESH_MODE_MIN_MID_MAX, u8HighThr is not used, else
    0<=u8LowThr<=u8HighThr<=255;*/
    MI_U8 u8MinVal; /*Minimum value when tri-level thresholding*/
    MI_U8 u8MidVal; /*Middle value when tri-level thresholding, if eMode<2, u32MidVal
    is not used; */
    MI_U8 u8MaxVal; /*Maxmum value when tri-level thresholding*/
}MI_IVE_ThreshCtrl_t;
```

➢ 成员

成员名称	描述
eMode	阈值化运算模式。
u8LowThresh	低阈值。 取值范围：[0,255]。
u8HighThresh	高阈值。 0≤u8LowThresh≤u8HighThresh≤255。
u8MinVal	最小值。 取值范围：[0,255]。
u8MidVal	中间值。 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

MI_IVE_ThreshMode_e

2.34. MI_IVE_SubMode_e

➢ 说明

定义两图像相减输出格式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_SUB_MODE_ABS = 0x0,        /*Absolute value of the difference*/
    E_MI_IVE_SUB_MODE_SHIFT = 0x1, /*The output result is obtained by shifting the result one digit right to reserve the signed bit. */
    E_MI_IVE_SUB_MODE_BUTT
}MI_IVE_SubMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_SUB_MODE_ABS	取差的绝对值。
E_MI_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

MI_IVE_SubCtrl_t

2.35. MI_IVE_SubCtrl_t

➢ 说明

定义两图像相减控制参数。

➢ 定义

```
typedef struct MI_IVE_SubCtrl_s
{
    MI_IVE_SubMode_e eMode;
}MI_IVE_SubCtrl_t;
```

➢ 成员

成员名称	描述
eMode	两图像相减模式

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_SubMode_e

2.36. MI_IVE_IntegOutCtrl_e

➢ 说明

定义积分图输出控制参数。

➢ 定义

```
typedef enum
{
    E_MI_IVE_INTEG_OUT_CTRL_COMBINE    = 0x0,
    E_MI_IVE_INTEG_OUT_CTRL_SUM
      = 0x1,
    E_MI_IVE_INTEG_OUT_CTRL_SQSUM
      = 0x2,
    E_MI_IVE_INTEG_OUT_CTRL_BUTT
}MI_IVE_IntegOutCtrl_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_INTEG_OUT_CTRL_COMBINE	和、平方和积分图组合输出，如图 1-13。
E_MI_IVE_INTEG_OUT_CTRL_SUM	仅和积分图输出。
E_MI_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

MI_IVE_IntegCtrl_t

2.37. MI_IVE_IntegCtrl_t

➢ 说明

定义积分图计算控制参数。

➢ 定义

```
typedef struct MI_IVE_IntegCtrl_s
{
    MI_IVE_IntegOutCtrl_e eOutCtrl;
}MI_IVE_IntegCtrl_t;
```

➢ 成员

成员名称	描述
eOutCtrl	积分图输出控制参数

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_IntegOutCtrl_e

2.38. MI_IVE_ThreshS16Mode_e

➢ 说明

定义 16bit 有符号图像的阈值化模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX = 0x0,
    E_MI_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX = 0x1,
    E_MI_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX = 0x2,
    E_MI_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX = 0x3,
    E_MI_IVE_THRESH_S16_MODE_BUTT
}MI_IVE_ThreshS16Mode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_THRESH_S16_MODE_S16_ TO_S8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = idVal; srcVal > highThr, dstVal = maxVal;
E_MI_IVE_THRESH_S16_MODE_S16_ TO_S8_MIN_ORI_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = rcVal; srcVal > highThr, dstVal = maxVal;
E_MI_IVE_THRESH_S16_MODE_S16_ TO_U8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = idVal; srcVal > highThr, dstVal = maxVal;
E_MI_IVE_THRESH_S16_MODE_S16_ TO_U8_MIN_ORI_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = rcVal; srcVal > highThr, dstVal = maxVal;

※ 注意事项

计算公式请参见 MI_IVE_ThreshS16 中的【注意】，示意图请参见图 2-9。

➢ 相关数据类型及接口

MI_IVE_ThreshS16Ctrl_t

2.39. MI_IVE_ThreshS16Ctrl_t

➢ 说明

定义 16bit 有符号图像的阈值化控制参数。

➢ 定义

```
typedef struct MI_IVE_ThreshS16Ctrl_s
{
    MI_IVE_ThreshS16Mode_e eMode;
    MI_S16 s16LowThr;          /*user-defined threshold*/
    MI_S16 s16HighThr;         /*user-defined threshold*/
    MI_IVE_Length8bit_u un8MinVal; /*Minimum value when tri-level thresholding*/
    MI_IVE_Length8bit_u un8MidVal; /*Middle value when tri-level thresholding*/
    MI_IVE_Length8bit_u un8MaxVal; /*Maxmum value when tri-level thresholding*/
}MI_IVE_ThreshS16Ctrl_t;
```

➢ 成员

成员名称	描述
eMode	阈值化运算模式。
s16LowThr	低阈值。
s16HighThr	高阈值。
un8MinVal	最小值。
un8MidVal	中间值。
un8MaxVal	最大值。

※ 注意事项

计算公式请参见 MI_IVE_ThreshS16 中的【注意】，示意图请参见图 2-9。

➢ 相关数据类型及接口

MI_IVE_ThreshS16Mode_e

2.40. MI_IVE_ThreshU16Mode_e

➢ 说明

定义 16bti 无符号图像的阈值化模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX = 0x0,
    E_MI_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX = 0x1, E_MI_IVE_THRESH_U16_MODE_BUTT
}MI_IVE_ThreshU16Mode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_THRESH_U16_MODE_U16_ TO_U8_MIN_MID_MAX	srcVal ≤ lowThr, dstVal = minVal; lowThr < srcVal ≤highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal;
E_MI_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

计算公式请参见 MI_IVE_ThreshU16 中的【注意】，示意图请参见图 2-10。

➢ 相关数据类型及接口

MI_IVE_ThreshU16Ctrl_t

2.41. MI_IVE_ThreshU16Ctrl_t

➢ 说明

定义 16bit 无符号图像的阈值化控制参数。

➢ 定义

```
typedef struct MI_IVE_ThreshU16Ctrl_s
{
    MI_IVE_ThreshU16Mode_e eMode;
    MI_U16 u16LowThr;
    MI_U16 u16HighThr;
    MI_U8 u8MinVal;
    MI_U8 u8MidVal;
    MI_U8 u8MaxVal;
}MI_IVE_ThreshU16Ctrl_t;
```

➢ 成员

成员名称	描述
eMode	阈值化运算模式。
u16LowThr	低阈值。
u16HighThr	高阈值。
u8MinVal	最小值。 取值范围：[0,255]。
u8MidVal	中间值。 取值范围：[0,255]。
u8MaxVal	最大值。 取值范围：[0,255]。

※ 注意事项

计算公式请参见 MI_IVE_ThreshU16 中的【注意】，示意图请参见图 2-10。

➢ 相关数据类型及接口

MI_IVE_ThreshU16Mode_e

2.42. MI_IVE_16BitTo8BitMode_e

➢ 说明

定义 16bit 图像数据到 8bit 图像数据的的转化模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8 = 0x0,
    E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS = 0x1,
    E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS = 0x2,
    E_MI_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8 = 0x3,
    E_MI_IVE_16BIT_TO_8BIT_MODE_BUTT
}MI_IVE_16BitTo8BitMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8	S16 数据到 S8 数据的线性变换。
E_MI_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。
E_MI_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。
E_MI_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8	S16 数据线性变换到 U8 数据。

※ 注意事项

计算公式请参见 MI_IVE_16BitTo8Bit 中的【注意】，示意图请参见图 2-11。

➢ 相关数据类型及接口

MI_IVE_16bitTo8BitCtrl_t

2.43. MI_IVE_16bitTo8BitCtrl_t

➢ 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 MI_IVE_16bitTo8BitCtrl_s
{
    MI_IVE_16BitTo8BitMode_e eMode;
    MI_U16 u16Denominator;
    MI_U8 u8Numerator;
    MI_S8 s8Bias;
}MI_IVE_16bitTo8BitCtrl_t;
```

➢ 成员

成员名称	描述
eMode	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

- 计算公式请参见 MI_IVE_ThreshU16 中的【注意】，示意图请参见图2-10。
- u8Numerator ≤u16Denominator，且u16Denominator≠0；

➢ 相关数据类型及接口

MI_IVE_16BitTo8BitMode_e

2.44. MI_IVE_OrdStatFilterMode_e

➢ 说明

定义顺序统计量滤波模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_ORD_STAT_FILTER_MODE_MEDIAN = 0x0,
    E_MI_IVE_ORD_STAT_FILTER_MODE_MAX = 0x1,
    E_MI_IVE_ORD_STAT_FILTER_MODE_MIN = 0x2,
    E_MI_IVE_ORD_STAT_FILTER_MODE_BUTT
}MI_IVE_OrdStatFilterMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_ORD_STAT_FILTER_MODE_MEDIAN	中值滤波。
E_MI_IVE_ORD_STAT_FILTER_MODE_MAX	最大值滤波，等价于灰度图的膨胀。
E_MI_IVE_ORD_STAT_FILTER_MODE_MIN	最小值滤波，等价于灰度图的腐蚀。

※ 注意事项

无。

➢ 相关数据类型及接口

IVE_ORD_STAT_FILTER_CTRL_S

2.45. MI_IVE_OrdStatFilter_t

➢ 说明

定义顺序统计量滤波控制参数。

➢ 定义

```
typedef struct MI_IVE_OrdStatFilter_s
{
    MI_IVE_OrdStatFilterMode_e eMode;
}MI_IVE_OrdStatFilter_t;
```

➢ 成员

成员名称	描述
eMode	顺序统计量滤波模式

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_OrdStatFilterMode_e

2.46. MI_IVE_MapLutMem_t

➢ 说明

定义 Map 算子的查找表内存信息。

➢ 定义

```
typedef struct MI_IVE_MapLutMem_s
{
    MI_U8 au8Map[MI_IVE_MAP_NUM];
}MI_IVE_MapLutMem_t;
```

➢ 成员

成员名称	描述
au8Map[MI_IVE_MAP_NUM]	Map 查找表数组。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.47. MI_IVE_EqualizeHistCtrlMem_t

➢ 说明

定义直方图均衡化辅助内存。

➢ 定义

```
typedef struct MI_IVE_EqualizeHistCtrlMem_s
{
    MI_U32
    au32Hist[MI_IVE_HIST_NUM];
    MI_U8
    au8Map[MI_IVE_MAP_NUM];
}MI_IVE_EqualizeHistCtrlMem_t;
```

➢ 成员

成员名称	描述
au32Hist[MI_IVE_HIST_NUM]	直方图统计的输出。
au8Map[MI_IVE_MAP_NUM]	根据统计直方图计算得到的 map 查找表。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_EqualizeHistCtrl_t

2.48. MI_IVE_EqualizeHistCtrl_t

➢ 说明

定义直方图均衡化控制参数。

➢ 定义

```
typedef struct MI_IVE_EqualizeHistCtrl_s
{
    MI_IVE_MemInfo_t stMem;
}MI_IVE_EqualizeHistCtrl_t;
```

➢ 成员

成员名称	描述
stMem	需开辟 sizeof(MI_IVE_EqualizeHistCtrlMem_t)字节大小的内存。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_EqualizeHistCtrlMem_t

2.49. MI_IVE_AddMode_e

➢ 说明

定义 add 计算模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_ADD_MODE_ROUNDING      =  0x0,
    E_MI_IVE_ADD_MOD_CLIPPING       =  0x1,

    E_MI_IVE_ADD_MODE_MAX
}MI_IVE_AddMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_ADD_MODE_ROUNDING	四舍五入模式
E_MI_IVE_ADD_MOD_CLIPPING	无条件舍弃模式

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE-AddCtrl_t

2.50. MI_IVE_AddCtrl_t

➢ 说明

定义两图像的加权加控制参数。

➢ 定义

```
typedef struct MI_IVE_AddCtrl_s
{
    MI_IVE_AddMode_e eMode;
    MI_U0Q16 u0q16X;          /*x of "xA+yB"*/
    MI_U0Q16 u0q16Y;          /*y of "xA+yB"*/
}MI_IVE_AddCtrl_t;
```

➢ 成员

成员名称	描述
eMode	计算模式
u0q16X	加权加“xA+yB”中的权重“x”。取值范围：[1, 65535]。
u0q16Y	加权加“xA+yB”中的权重“y”。取值范围：{65536 – u0q16X}。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_AddMode_e

2.51. MI_IVE_NccDstMem_t

➢ 说明

定义 NCC 的输出内存信息。

➢ 定义

```
typedef struct MI_IVE_NccDstMem_s
{
    MI_U64 u64Numerator;
    MI_U64 u64QuadSum1;
    MI_U64 u64QuadSum2;
}MI_IVE_NccDstMem_t;
```

➢ 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： $\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src1}(i,j))$
u64QuadSum2	NCC 计算公式的分母--根号内部分：

成员名称	描述
	$\sum_{i=1}^{w}\sum_{j=1}^{h}(I^{2}_{src2}(i,j))$

※ 注意事项

计算公式请参见 MI_IVE_Ncc 中的【注意】。

➢ 相关数据类型及接口

无。

2.52. MI_IVE_Region_t

➢ 说明

定义连通区域信息。

➢ 定义

```
typedef struct MI_IVE_Region_s
{
    MI_U32 u32Area;          /*Represented by the pixel number*/
    MI_U16 u16Left;          /*Circumscribed rectangle left border*/
    MI_U16 u16Right;         /*Circumscribed rectangle right border*/
    MI_U16 u16Top;           /*Circumscribed rectangle top border*/
    MI_U16 u16Bottom;        /*Circumscribed rectangle bottom border*/
}MI_IVE_Region_t;
```

➢ 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

MI_IVE_CcBlob_t

2.53. MI_IVE_CcBlob_t

➢ 说明

定义连通区域标记的输出信息。

➢ 定义

```
typedef struct MI_IVE_CcBlob_s
{
    MI_U16 u16CurAreaThr; /*Threshold of the result regions' area*/
    MI_S8 s8LabelStatus; /*-1: Labeled failed ; 0: Labeled successfully*/
    MI_U8 u8RegionNum; /*Number of valid region, non-continuous stored*/
    MI_IVE_Region_t astRegion[MI_IVE_MAX_REGION_NUM]; /*Valid regions with 'u32Area>0' and 'label = ArrayIndex+1'*/
}MI_IVE_CcBlob_t;
```

➢ 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[MI_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。其中索引254 记录因pstCclCtrl→u16InitAreaThr 而被删除的连通区域总面积。

※ 注意事项

无。

※ 注意事项

MI_IVE_Region_t

2.54. MI_IVE_CclMode_e

➢ 说明

定义连通区域模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_CCL_MODE_4C        =       0x0,
      /*4-connectivity*/
    E_MI_IVE_CCL_MODE_8C        =       0x1,
      /*8-connectivity*/
    E_MI_IVE_CCL_MODE_BUTT
}MI_IVE_CclMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_CCL_MODE_4C	4-连通。
E_MI_IVE_CCL_MODE_8C	8-连通。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.55. MI_IVE_CclCtrl_t

➢ 说明

定义连通区域标记控制参数。

➢ 定义

```
typedef struct MI_IVE_CclCtrl_s
{
    MI_U16 u16InitAreaThr;      /*Init threshold of region area*/
    MI_U16 u16Step;             /*Increase area step for once*/
}MI_IVE_CclCtrl_t;
```

➢ 成员

成员名称	描述
u16InitAreaThr	初始面积阈值。 取值范围：[0, 65535]。参考取值：4。
u16Step	面积阈值增长步长。取值范围：[1,65535]。参考取值：2。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_IVE_CcBlob_t

2.56. MI_IVE_GmmCtrl_t

➢ 说明

定义 GMM 背景建模的控制参数。

➢ 定义

```
typedef struct MI_IVE_GmmCtrl_s
{
    MI_U22Q10    u22q10NoiseVar;        /*Initial noise Variance*/
    MI_U22Q10    u22q10MaxVar;          /*Max  Variance*/
    MI_U22Q10    u22q10MinVar;          /*Min  Variance*/
    MI_U0Q16     u0q16LearnRate;        /*Learning rate*/
    MI_U0Q16     u0q16BgRatio;          /*Background ratio*/
    MI_U8Q8      u8q8VarThr;            /*Variance Threshold*/
    MI_U0Q16     u0q16InitWeight;       /*Initial Weight*/
    MI_U8        u8ModelNum;            /*Model number: 3 or 5*/
}MI_IVE_GmmCtrl_t;
```

➢ 成员

成员名称	描述
u22q10NoiseVar	初始噪声方差。 取值范围：[0x1，0xFFFFFF]。 对灰度的 GMM，对应 OpenCV MOG 中灰度模型中的noiseSigma * noiseSigma。 参考取值：15*15*(1<<10)。 对RGB 的GMM，对应 OpenCV MOG 中RGB 模型中的 3 * noiseSigma * noiseSigma 。 参考取值：3*15*15*(1<<10)。
u22q10MaxVar	模型方差的最大值。 取值范围；[0x1，0xFFFFFF]。 对应 OpenCV MOG2 中 fVarMax。 参考取值：3*4000<<10（RGB），2000<<10（灰度）。

成员名称	描述
u22q10MinVar	模型方差的最小值。 取值范围：[0x1，22q10MaxVar]。 对应 OpenCV MOG2 中 fVarMin。 参考取值：600<<10（RGB），200<<10（灰度）。
u0q16LearnRate	学习速率。 取值范围：[1，65535]。 对应OpenCV MOG2 中learningRate。 参考取值： if（frameNum<500）(1/frameNum)*((1<<16)-1)； else ((1/500)*((1<<16)-1)。
u0q16BgRatio	背景比例阈值。 取值范围：[1，65535]。 对应OpenCV MOG 中backgroundRatio。 参考取值：0.8*((1<<16)-1)。
u8q8VarThr	方差阈值。 取值范围：[1，65535]。 对应OpenCV MOG 中varThreshold，用于决定一个像素是否命中当前模型。 参考取值：6.25*(1<<8)。
u0q16InitWeight	初始权重。 取值范围：[1，65535]。 对应OpenCV MOG 中的defaultInitialWeight。 参考取值：0.05*((1<<16)-1)。
u8ModelNum	模型个数。 取值范围：{3,5}。 对应 OpenCV MOG 中 nmixtures。

※ 注意事项

无。

➢ 相关数据类型及接口

无

2.57. MI_IVE_CannyStackSize_t

➢ 说明

定义 Canny 边缘前半部分计算时强边缘点栈大小结构体。

➢ 定义

```
typedef struct MI_IVE_CannyStackSize_s
{
    MI_U32 u32StackSize;                    /*Stack size for output*/
    MI_U8 u8Reserved[MI_IVE_CANNY_STACK_RESERVED_SIZE];    /*For 16 byte
    align*/
}MI_IVE_CannyStackSize_t;
```

➢ 成员

成员名称	描述
u32StackSize	栈大小(强边缘点的个数)。
u8Reserved[MI_IVE_CANNY_STACK_RESERVED_SIZE]	保留位。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.58. MI_IVE_CannyHysEdgeCtrl_t

➢ 说明

定义 Canny 边缘前半部分计算任务的控制参数。

➢ 定义

```
typedef struct MI_IVE_CannyHysEdgeCtrl_s
{
    MI_IVE_MemInfo_t
    stMem;
    MI_U16 u16LowThr;
    MI_U16 u16HighThr;
    MI_S8
    as8Mask[MI_IVE_MASK_
    SIZE_5X5];
} MI_IVE_CannyHysEdgeCtrl_t;
```

➢ 成员

成员名称	描述
stMem	辅助内存。内存配置大小说明见 MI_IVE_CannyHysEdge 的【注意】。
u16LowThr	低阈值。 取值范围：[0,255]。
u16HighThr	高阈值。 取值范围：[u16LowThr,255]。
as8Mask[MI_IVE_MASK_SIZE_5X5]	用于计算梯度的参数模板。

※ 注意事项

无。

➢ 相关数据类型及接口

无。

2.59. MI_IVE_LbpCmpMode_e

➢ 说明

定义 LBP 计算的比较模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_LBP_CMP_MODE_NORMAL = 0x0, /* P(x)-P(center)>= un8BitThr.s8Val, s(x)=1; else s(x)=0; */
    E_MI_IVE_LBP_CMP_MODE_ABS = 0x1, /* abs(P(x)-P(center))>=un8BitThr.u8Val, s(x)=1; else s(x)=0; */
    E_MI_IVE_LBP_CMP_MODE_ABS_MUL = 0x2,
    E_MI_IVE_LBP_CMP_MODE_MAX
}MI_IVE_LbpCmpMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_LBP_CMP_MODE_NORMAL	LBP 简单比较模式。
E_MI_IVE_LBP_CMP_MODE_ABS	LBP 绝对值比较模式。
E_MI_IVE_LBP_CMP_MODE_MAX	LBP 绝对值乘法比较模式

※ 注意事项

计算公式参考 MI_IVE_Lbp 中的【注意】，示意图请参考图 2-16。

➢ 相关数据类型及接口

MI_IVE_LbpCtrl_t

2.60. MI_IVE_LbpChalMode_e

➢ 说明

定义 LBP 输入信道模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_LBP_CHAL_MODE_U8C1     = 0x0,
    E_MI_IVE_LBP_CHAL_MODE_U8C2     = 0x1,

    E_MI_IVE_LBP_CHAL_MODE_MAX
} MI_IVE_LbpChalMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_LBP_CHAL_MODE_U8C1	只有单一输入图像模式。
E_MI_IVE_LBP_CHAL_MODE_U8C2	两个输入影像模式。

※ 注意事项

计算公式参考 MI_IVE_Lbp 中的【注意】，示意图请参考图 2-16。

➢ 相关数据类型及接口

MI_IVE_LbpCtrl_t

2.61. MI_IVE_LbpCtrl_t

➢ 说明

定义 LBP 纹理计算控制参数。

➢ 定义

```
typedef struct MI_IVE_LbpCtrl_s
{
    MI_IVE_LbpCmpMode_e
    eMode;
    MI_IVE_Length8bit_u un8BitThr;
}MI_IVE_LbpCtrl_t;
```

➢ 成员

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

※ 注意事项

计算公式参考 MI_IVE_Lbp 中的【注意】，示意图请参考图 2-16。

➢ 相关数据类型及接口

- MI_IVE_LbpCmpMode_e
- MI_IVE_LbpChalMode_e
- MI_IVE_Length8bit_u

2.62. MI_IVE_NormGradOutCtrl_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
{
    E_MI_IVE_NORM_GRAD_OUT_CTRL_HOR_AND_
    VER   = 0x0,
    E_MI_IVE_NORM_GRAD_OUT_CTRL_HOR
      =  0x1,
    E_MI_IVE_NORM_GRAD_OUT_CTRL_VER
      = 0x2,
    E_MI_IVE_NORM_GRAD_OUT_CTRL_COMBINE
          = 0x3,
    E_MI_IVE_NORM_GRAD_OUT_CTRL_BUTT
}MI_IVE_NormGradOutCtrl_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_NORM_GRAD_OUT_CTRL_ HOR_AND_VER	同时输出梯度信息的 H、V 分量图(H、V 定义见 MI_ IVE_NormGrad 的【参数】)。
E_MI_IVE_NORM_GRAD_OUT_CTRL_ HOR	仅输出梯度信息的 H 分量图。
E_MI_IVE_NORM_GRAD_OUT_CTRL_ VER	仅输出梯度信息的 V 分量图。
E_MI_IVE_NORM_GRAD_OUT_CTRL_ COMBINE	输出梯度信息以 package 存储（如图1-7）的 HV 图。

※ 注意事项

无。

➢ 相关数据类型及接口

IVE_NORM_GRAD_CTRL_S

2.63. MI_IVE_NormGradCtrl_t

➢ 说明

定义归一化梯度信息计算控制参数。

➢ 定义

```
typedef struct MI_IVE_NormGradCtrl_s
{
    MI_IVE_NormGradOutCtrl_e eOutCtrl;
    MI_S8
    as8Mask[MI_IVE_MASK_SIZE_5X5];
    MI_U8 u8Norm;
}MI_IVE_NormGradCtrl_t;
```

➢ 成员

成员名称	描述
eOutCtrl	梯度信息输出控制模式。
as8Mask[MI_IVE_MASK_SIZE_5X5]	计算梯度需要的模板。
u8Norm	归一化参数。 取值范围：[1,13]。

※ 注意事项

无。

➢ 相关数据类型及接口

E_MI_IVE_NORM_GRAD_OUT_CTRL_E

2.64. MI_IVE_MvS9Q7_t

➢ 说明

定义 LK 光流位移结构体。

➢ 说明

```
typedef struct MI_IVE_MvS9Q7_s
{
    MI_S32      s32Status;      /*Result of tracking: 0-success; -1-failure*/
    MI_S9Q7     s9q7Dx;         /*X-direction component of the movement*/
    MI_S9Q7     s9q7Dy;         /*Y-direction component of the movement*/
}MI_IVE_MvS9Q7_t;
```

➢ 成员

成员名称	描述
s32Status	特征点跟踪的状态。 0：成功； 1：失败。
s9q7Dx	特征点位移的 x 分量。
s9q7Dy	特征点位移的 y 分量。

※ 注意事项

无。

➢ 相关数据类型及接口

无

2.65. MI_IVE_LkOpticalFlowCtrl_t

➢ 说明

定义 LK 光流计算控制参数。

➢ 定义

```
typedef struct MI_IVE_LkOpticalFlowCtrl_s
{
    MI_U16 u16CornerNum;      /*Number of the feature points, <200*/
    MI_U0Q8 u0q8MinEigThr;    /*Minimum eigenvalue threshold*/
    MI_U8 u8IterCount;        /*Maximum iteration times*/
    MI_U0Q8 u0q8Epsilon;      /*Threshold of iteration for dx^2 + dy^2 < u0q8Epsilon
*/
}MI_IVE_LkOpticalFlowCtrl_t;
```

➢ 成员

成员名称	描述
u16CornerNum	输入的角点\特征点数目。 取值范围：[1,200]。
u0q8MinEigThr	最小特征值阈值。取 值范围：[1,255]。
u8IterCount	最大迭代次数。 取 值范围：[1,20]。
u0q8Epsilon	迭代收敛条件：dx^2 + dy^2 < u0q8Epsilon。 取值范围：[1,255]。 参考取值：2。

※ 注意事项

无

➢ 相关数据类型及接口

无

2.66. MI_IVE_SadMode_e

➢ 说明

定义 SAD 计算模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_SAD_MODE_MB_4X4 = 0x0,
    /*4x4*/
    E_MI_IVE_SAD_MODE_MB_8X8 = 0x1,
    /*8x8*/
    E_MI_IVE_SAD_MODE_MB_16X16     = 0x2,
    /*16x16*/
    E_MI_IVE_SAD_MODE_BUTT
}MI_IVE_SadMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_SAD_MODE_MB_4X4	按 4x4 像素块计算 SAD。
E_MI_IVE_SAD_MODE_MB_8X8	按 8x8 像素块计算 SAD。
E_MI_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

MI_IVE_SadCtrl_t

2.67. MI_IVE_SadOutCtrl_e

➢ 说明

定义 SAD 输出控制模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_SAD_OUT_CTRL_16BIT_BOTH   = 0x0,  /*Output 16 bit sad and
    thresh*/
    E_MI_IVE_SAD_OUT_CTRL_8BIT_BOTH    = 0x1,  /*Output 8 bit sad and
    thresh*/
    E_MI_IVE_SAD_OUT_CTRL_16BIT_SAD    = 0x2,  /*Output 16 bit sad*/
    E_MI_IVE_SAD_OUT_CTRL_8BIT_SAD     = 0x3,  /*Output 8 bit sad*/
    E_MI_IVE_SAD_OUT_CTRL_THRESH  = 0x4,  /*Output thresh, 16 bits sad */
    E_MI_IVE_SAD_OUT_CTRL_BUTT
}MI_IVE_SadOutCtrl_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_SAD_OUT_CTRL_16BIT_BOTH	16 bit SAD 图和阈值化图输出模式。
E_MI_IVE_SAD_OUT_CTRL_8BIT_BOTH	8 bit SAD 图和阈值化图输出模式。
E_MI_IVE_SAD_OUT_CTRL_16BIT_SAD	16 bit SAD 图输出模式。
E_MI_IVE_SAD_OUT_CTRL_8BIT_SAD	8 bit SAD 图输出模式。
E_MI_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

MI_IVE_SadCtrl_t

2.68. MI_IVE_SadCtrl_t

➢ 说明

定义 SAD 控制参数。

➢ 定义

```
typedef struct MI_IVE_SadCtrl_s
{
    MI_IVE_SadMode_e eMode;
    MI_IVE_SadOutCtrl_e eOutCtrl;
    MI_U16 u16Thr; /*srcVal <= u16Thr, dstVal = minVal;srcVal > u16Thr, dstVal = maxVal.*/
    MI_U8 u8MinVal; /*Min value*/
    MI_U8 u8MaxVal; /*Max value*/
}MI_IVE_SadCtrl_t;
```

➢ 成员

成员名称	描述
eMode	SAD 计算模式。
eOutCtrl	SAD 输出控制模式。
u16Thr	对计算的 SAD 图进行阈值化的阈值。
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

- MI_IVE_SadMode_e
- MI_IVE_SadOutCtrl_e

2.69. MI_IVE_BernsenCtrl_t

➢ 说明

定义 Bernsen 控制参数。

➢ 定义

```
typedef struct MVE_IVE_BernsenCtrl_s
{
    MI_IVE_BernsenMode_e enMode;
    MI_U8 u8WinSize;
    MI_U8 u8MaxVal;
} MVE_IVE_BernsenCtrl_t;
```

➢ 成员

成员名称	描述
enMode	Bernsen 阀值模式。
u8WinSize	窗口大小。范围：{3, 5}
u8MaxVal	模式 MVE_BERNSEN_MODE_THRESH 的门坎值。范围：[0, 255]

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_BernsenMode_e

2.70. MI_IVE_BernsenMode_e

➢ 说明

定义 Bernsen 阀值模式。

➢ 定义

```
typedef enum
{
        E_MI_IVE_BERNSEN_MODE_NORMAL = 0x0,
        E_MI_IVE_BERNSEN_MODE_THRESH = 0x1,

        E_MI_IVE_BERNSEN_MODE_MAX
} MVE_IVE_BernsenMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_BERNSEN_MODE_NORMA	简单 Bernsen 阀值模式
E_MI_IVE_BERNSEN_MODE_THRESH	基于全局阀值和本地 Bernsen 阀值的阀值模式
E_MI_IVE_ERNSEN_MODE_MAX	错误模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_BernsenCtrl_t

2.71. MI_IVE_LineFilterHorCtrl_t

➢ 说明

定义 LineFilterHor 控制参数。

➢ 定义

```
typedef struct MVE_IVE_LineFilterHorCtrl_s
{
        MI_U8 u8GapMinLen;
        MI_U8 u8DensityThr;
        MI_U8 u8HorThr;
} MVE_IVE_LineFilterHorCtrl_t;
```

➢ 成员

成员名称	描述
u8GapMinLen	最小线段长度，详情请看 MI_IVE_LineFilterHor 中的 thr1 范围 ：[1, 20]

成员名称	描述
u8DensityThr	密度阀值，详情请看 MI_IVE_LineFilterHor 中的 thr2 范围 ：[1, 50]
u8HorThr	水平线段长度阀值，详情请看 MI_IVE_LineFilterHor 中的 thr3 范围 ：[1, 50]

※ 注意事项

无

➢ 相关数据类型及接口

无

2.72. MI_IVE_LineFilterVerCtrl_t

➢ 说明

定义 LineFilterVer 控制参数。

➢ 定义

```
typedef struct MVE_IVE_LineFilterVerCtrl_s
{
        MI_U8 u8VerThr;
} MVE_IVE_LineFilterVerCtrl_t _t;
```

➢ 成员

成员名称	描述
u8VerThr	垂直线段长度阀值，详情请看 MI_IVE_LineFilterVer 中的 thr 范围 ：[1, 64]

※ 注意事项

无

➢ 相关数据类型及接口

无

2.73. MI_IVE_NoiseRemoveHor_t

➢ 说明

定义 NoiseRemoveHor 控制参数。

➢ 定义

```
typedef struct MVE_IVE_NoiseRemoveHorCtrl_s
{
    MI_U8 u8HorThr;
    MI_U8 u8HorThrMax;
} MVE_IVE_NoiseRemoveHorCtrl_t;
```

➢ 成员

成员名称	描述
u8HorThr	决定水平噪声的线段长度阀值，详情请看 MI_IVE_NoiseRemoveHor 中的 thr1 范围 ：[1, 100]
u8HorThrMax	决定水平噪声的最大线段长度阀值，详情请看 MI_IVE_NoiseRemoveHor 中的 thr2 范围 ：[1, 100]

※ 注意事项

u8HorThrMax 一定大于 u8HorThr。

➢ 相关数据类型及接口

无

2.74. MI_IVE_NoiseRemoveVer_t

➢ 说明

定义 NoiseRemoveVer 控制参数。

➢ 定义

```
typedef struct MVE_IVE_NoiseRemoveVerCtrl_s
{
    MI_U8 u8VerThr;
    MI_U8 u8VerThrMax;
} MVE_IVE_NoiseRemoveVerCtrl_t;
```

➢ 成员

成员名称	描述
u8VerThr	决定垂直噪声的线段长度阀值，详情请看 MI_IVE_NoiseRemoveVer 中的 thr1 范围 ：[1, 100]
u8VerThrMax	决定垂直噪声的最大线段长度阀值，详情请看 MI_IVE_NoiseRemoveVer 中的 thr2 范围 ：[1, 100]

※ 注意事项

u8VerThrMax一定大于 u8VerThr。

➢ 相关数据类型及接口

无

2.75. MI_IVE_AdpThreshCtrl_t

➢ 说明

定义 AdpThresh 控制参数。

➢ 定义

```
typedef struct MVE_IVE_AdpThreshCtrl_s
{
    MI_U8 u8RateThr;
    MI_U8 u8HalfMaskx;
    MI_U8 u8HalfMasky;
    MI_U8 s16Offset;
    MI_U8 u8ValueThr;
} MVE_IVE_AdpThreshCtrl_t;
```

➢ 成员

成员名称	描述
u8RateThr	阀值比例，详情请看 MI_IVE_AdpThresh 中的 RateThr 范围: [1, 20]
u8HalfMaskx	窗口宽的一半，详情请看 MI_IVE_AdpThresh 中的 w 范围: [1, 40]

成员名称	描述
u8HalfMasky	窗口高的一半，详情请看 MI_IVE_AdpThresh 中的 h 范围: [1, 40]
s16Offset	偏移量，详情请看 MI_IVE_AdpThresh 中的 offset 范围: [-128, 127]
u8ValueThr	像素阀值，详情请看 MI_IVE_AdpThresh 中的 ValueThr 范围: [1, 255]

※ 注意事项

无

➢ 相关数据类型及接口

无

2.76. MI_IVE_ResizeCtrl_t

➢ 说明

定义 Resize 控制参数。

➢ 定义

```
typedef struct _MVE_IVE_ResizeCtrl_s
{
    MVE_IVE_ResizeMode_e enMode;
} MVE_IVE_ResizeCtrl_t;
```

➢ 成员

成员名称	描述
enMode	输入图像模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_ResizeMode_e

2.77. MI_IVE_ResizeMode_e

➢ 说明

定义 Resize 输入模式。

➢ 定义

```
typedef enum
{
        E_MI_IVE_RESIZE_TYPE_U8C1              = 0x0,
        E_MI_IVE_RESIZE_TYPE_U8C3_PLANAR       = 0x1,
        E_MI_IVE_RESIZE_TYPE_U8C3_PACKAGE      = 0x2,
        E_MI_IVE_RESIZE_TYPE_YUV420SP          = 0x3,

        E_MI_IVE_RESIZE_TYPE_MAX
} MVE_IVE_ResizeMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_RESIZE_TYPE_U8C1	输入图像的每个像素用 1 个 8bit 无符号数据表示单通道图像
E_MI_IVE_RESIZE_TYPE_U8C3_PLANAR	输入图像的每个像素用 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。
E_MI_IVE_RESIZE_TYPE_U8C3_PACKAGE	输入图像的每个像素用 3 个 8bit 无符号数据表示且以 package 格式存储 3 通道图像。
E_MI_IVE_RESIZE_TYPE_YUV420SP	输入为 YUV420 Semiplanar 格式的图像。
E_MI_IVE_RESIZE_TYPE_MAX	错误模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_ResizeCtrl_t

2.78. MI_IVE_BatCtrl_t

➢ 说明

定义 BAT 控制参数。

➢ 定义

```
typedef struct _MVE_IVE_BatCtrl_s
{
    MVE_IVE_BatMode_e enMode;
    MI_U16 _t u16HorTimes;
    MI_U16 _t u16VerTimes;
} MVE_IVE_BatCtrl_t;
```

➢ 成员

成员名称	描述
enMode	输入图像模式
u16HorTimes	水平侦测阀值
u16VerTimes	垂直侦测阀值

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_BatMode_e

2.79. MI_IVE_BatMode_e

➢ 说明

定义 Bat 运算模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_BAT_OUT_CTRL_BOTH = 0x0,
    E_MI_IVE_BAT_OUT_CTRL_HOR  = 0x1,
    E_MI_IVE_BAT_OUT_CTRL_VER  = 0x2,

    E_MI_IVE_BAT_OUT_CTRL_MAX
} MVE_IVE_BatMode_e
```

➢ 成员

成员名称	描述
E_MI_IVE_BAT_OUT_CTRL_BOTH	水平和垂直共同模式
E_MI_IVE_BAT_OUT_CTRL_HOR	水平模式
E_MI_IVE_BAT_OUT_CTRL_VER	垂直模式
E_MI_IVE_BAT_OUT_CTRL_MAX	错误模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_BatCtrl_t

2.80. MI_IVE_AccCtrl_t

➢ 说明

定义 Acc 控制参数。

➢ 定义

```
typedef struct MVE_IVE_AccCtrl_s
{
    MVE_IVE_AccMode_e enMode;
} MVE_IVE_AccCtrl_t;
```

➢ 成员

成员名称	描述
enMode	输入图像模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_AccMode_e

2.81. MI_IVE_AccMode_e

➢ 说明

定义 Acc 运算模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_ACC_MODE_INCREASE              = 0x0,
    E_MI_IVE_ACC_MODE_DECREASE              = 0x1,
    E_MI_IVE_ACC_MODE_INCREASE_MAP_255TO1   = 0x2,

    E_MI_IVE_ACC_MODE_MAX
} MVE_IVE_AccMode_e
```

➢ 成员

成员名称	描述
E_MI_IVE_ACC_MODE_INCREASE	累积增加模式
E_MI_IVE_ACC_MODE_DECREASE	累积减少模式
E_MI_IVE_ACC_MODE_INCREASE_MAP_255TO1	累计增加，且 255 转换为 1 模式
E_MI_IVE_BAT_OUT_CTRL_MAX	错误模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_AccCtrl_t

2.82. MI_IVE_MatrTranfMode_e

➢ 说明

定义 matrix_transform 的输入信道模式。

➢ 定义

```
typedef enum
{
    E_MI_IVE_MATRIX_TRANSFORM_TYPE_C1      = 0x0,
    E_MI_IVE_MATRIX_TRANSFORM_TYPE_C2      = 0x1,
    E_MI_IVE_MATRIX_TRANSFORM_TYPE_C3      = 0x2,

    E_MI_IVE_MATRIX_TRANSFORM_TYPE_MAX
}MVE_IVE_MatrTranfMode_e;
```

➢ 成员

成员名称	描述
E_MI_IVE_MATRIX_TRANSFORM_TYPE_C1	单一输入图像模式
E_MI_IVE_MATRIX_TRANSFORM_TYPE_C2	两个输入图像模式
E_MI_IVE_MATRIX_TRANSFORM_TYPE_C3	三个输入图像模式
E_MI_IVE_MATRIX_TRANSFORM_TYPE_MAX	错误模式

※ 注意事项

无

➢ 相关数据类型及接口

- MI_IVE_MatrTranfCtrl_t

2.83. MI_IVE_MatrTranfCtrl_t

➢ 说明

定义 matrix_transform 控制参数。

➢ 定义

```
typedef struct MI_IVE_MatrTranfCtrl_S
{
    MVE_IVE_MatrTranfMode_e enMode;      /*Input channel mode*/
    MI_S16 s16MatrixArray[9];   //Official
} MI_IVE_MatrTranfCtrl_t;
```

➢ 成员

成员名称	描述
enMode	输入信道模式
s16MatrixArray[9]	矩阵系数

※ 注意事项

若 enMode = E_MI_IVE_MATRIX_TRANSFORM_TYPE_C1，则只需输入 s16MatrixArray[0] 即可

若 enMode = E_MI_IVE_MATRIX_TRANSFORM_TYPE_C2，则只需输入 s16MatrixArray[0] ~ s16MatrixArray[3] 即可

若 enMode = E_MI_IVE_MATRIX_TRANSFORM_TYPE_C23，则全部矩阵系数都需输入

➢ 相关数据类型及接口

- MI_IVE_MatrTranfMode_e

3. IVE 错误码

智能加速引擎 API 错误码如表 3-1 所示。

表 3-1 智能加速引擎 API 错误码

错误代码	宏定义	描述
0xA01D8001	MI_ERR_IVE_INVALID_DEVID	设备 ID 超出合法范围
0xA01D8002	MI_ERR_IVE_INVALID_CHNID	通道组号错误或无效区域句柄
0xA01D8003	MI_ERR_IVE_ILLEGAL_PARAM	参数超出合法范围
0xA01D8004	MI_ERR_IVE_EXIST	重复执行已存在的设备、通道或资源
0xA01D8005	MI_ERR_IVE_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
0xA01D8006	MI_ERR_IVE_NULL_PTR	函数参数中有空指针
0xA01D8007	MI_ERR_IVE_NOT_CONFIG	模块没有配置
0xA01D8008	MI_ERR_IVE_NOT_SUPPORT	不支持的参数或者功能
0xA01D8009	MI_ERR_IVE_NOT_PERM	该操作不允许，如试图修改静态配置参数
0xA01D800C	MI_ERR_IVE_NOMEM	分配内存失败，如系统内存不足
0xA01D800D	MI_ERR_IVE_NOBUF	分配缓存失败，如申请的图像缓冲区太大
0xA01D800E	MI_ERR_IVE_BUF_EMPTY	缓冲区中无图像
0xA01D800F	MI_ERR_IVE_BUF_FULL	缓冲区中图像满
0xA01D8010	MI_ERR_IVE_NOTREADY	系统没有初始化或没有加载相应模块
0xA01D8011	MI_ERR_IVE_BADADDR	地址非法
0xA01D8012	MI_ERR_IVE_BUSY	系统忙
0xA01D8040	MI_ERR_IVE_SYS_TIMEOUT	系统超时
0xA01D8041	MI_ERR_IVE_QUERY_TIMEOUT	Query 查询超时
0xA01D8042	MI_ERR_IVE_OPEN_FILE	打开文件失败
0xA01D8043	MI_ERR_IVE_READ_FILE	读文件失败
0xA01D8044	MI_ERR_IVE_WRITE_FILE	写文件失败