

IVE

API 参考

文档版本 V1.4

发布日期 2025-08-20

前言

概述

本文档为使用媒体处理芯片的 IVE 协处理器进行智能分析方案开发的程序员而写，目的是供您在开发过程中查阅 IVE 协处理器支持的各种参考信息，包括 API、头文件、错误码、Proc 信息等。



说明

未经特殊说明，以 XM7606V13 指代 XM7606 系列所有芯片，以 XM7206V11A 指代 XM7206 系列所有芯片。

产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	表示如不可避免则将会导致死亡或严重伤害的具有高等级风险的危害。
 警告	表示如不可避免则可能导致死亡或严重伤害的具有中等级风险的危害。
 注意	表示如不可避免则可能导致轻微或中度伤害的具有低等级风险的危害。
 须知	用于传递设备或环境安全警示信息。如不可避免则可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “须知”不涉及人身伤害。
 说明	对正文中重点信息的补充说明。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2024-12-21	V1.0	DI 版本发布。
2025-03-04	V1.1	1.3 改动点 第三条加入 STCandiCorner 也不支持 1920*1080 的分辨率。
2025-04-15	V1.2	“1 概述”更新概述和功能描述，删除“1.2 整体架构”，删除“1.3 改动点”，新增“1.2 功能描述”。 “2 驱动设计”删除。 “3 UMD API 参考”调整为“2 API 参考”、“3 数据类型和数据结构”、“5 Proc 调试信息”三个章节。

修订日期	版本	修订说明
		新增 “4 错误码” 。
2025-06-05	V1.3	3 数据类型和数据结构 xmedia_ive_st_corner_ctrl_s 结构体中 max_corner_num 取值范围修正为[1, 500]。
		2 API 参考 新增 xmedia_ive_lbp 新增 xmedia_ive_acc 新增 xmedia_ive_lk_optical_flow_pyr 新增 xmedia_ive_and_ccl 新增 xmedia_ive_or_ccl 新增 xmedia_ive_xor_ccl 新增 xmedia_ive_sub_ccl 新增 xmedia_ive_add_ccl 新增 xmedia_ive_sub_hist 新增 xmedia_ive_sub_thresh 新增 xmedia_ive_sub_integ 新增 xmedia_ive_sub_map 新增 xmedia_ive_add_hist 新增 xmedia_ive_add_thresh 新增 xmedia_ive_add_integ 新增 xmedia_ive_add_map 新增 xmedia_ive_filter_thresh 新增 xmedia_ive_ord_stat_filter_thresh 新增 xmedia_ive_sobel_thresh_s16 新增 xmedia_ive_mag_and_ang_thresh_u16 新增 xmedia_ive_mag_and_ang_16bit_to_8bit 新增 xmedia_ive_thresh_dilate 新增 xmedia_ive_thresh_erode

修订日期	版本	修订说明
2025-08-20	V1.4	“1.2.1 重要概念” 更新。 “3 数据类型和数据结构” 更新。

目 录

前 言.....	i
1 概述.....	1
1.1 概述.....	1
1.2 功能描述.....	1
1.2.1 重要概念.....	1
1.2.2 使用示意.....	22
2 API 参考.....	23
3 数据类型和数据结构.....	153
4 错误码.....	250
4.1 IVE 错误码.....	250
5 Proc 调试信息.....	253
5.1 概述.....	253
5.2 Proc 信息说明.....	253

插图目录

图 1-1 跨度 (stride) 示意图.....	2
图 1-2 XMEDIA_IVE_IMAGE_TYPE_U8C1 \ XMEDIA_IVE_IMAGE_TYPE_S8C1 \ XMEDIA_IVE_IMAGE_TYPE_S16C1 \ XMEDIA_IVE_IMAGE_TYPE_U16C1 \ XMEDIA_IVE_IMAGE_TYPE_S32C1 \ XMEDIA_IVE_IMAGE_TYPE_U32C1 \ XMEDIA_IVE_IMAGE_TYPE_S64C1 \ XMEDIA_IVE_IMAGE_TYPE_U64C1 类型的 xmedia_ive_image_s 图像	6
图 1-3 XMEDIA_IVE_IMAGE_TYPE_YUV420SP 类型的 xmedia_ive_image_s 图像.....	7
图 1-4 XMEDIA_IVE_IMAGE_TYPE_YUV422SP 类型的 xmedia_ive_image_s 图像.....	7
图 1-5 XMEDIA_IVE_IMAGE_TYPE_YUV420P 类型的 xmedia_ive_image_s 图像	8
图 1-6 XMEDIA_IVE_IMAGE_TYPE_YUV422P 类型的 xmedia_ive_image_s 图像	9
图 1-7 XMEDIA_IVE_IMAGE_TYPE_S8C2_PACKAGE 类型的 xmedia_ive_image_s 图像.....	9
图 1-8 XMEDIA_IVE_IMAGE_TYPE_S8C2_PLANAR 类型的 xmedia_ive_image_s 图像	10
图 1-9 XMEDIA_IVE_IMAGE_TYPE_U8C3_PACKAGE 类型的 xmedia_ive_image_s 图像	10
图 1-10 XMEDIA_IVE_IMAGE_TYPE_U8C3_PLANAR 类型的 xmedia_ive_image_s 图像	11
图 1-11 xmedia_ive_data_s 类型的数据内存示意	12
图 1-12 xmedia_ive_mem_s 类型的数据内存示意	12
图 1-13 积分图 (XMEDIA_IVE_IMAGE_TYPE_U64C1) 组合输出示意	12
图 1-14 直方图输出格式示意	12
图 1-15 XMEDIA_IVE_BLOB_TYPE_S32 类型 xmedia_ive_blob_s (2 通道 2 帧示意图)	15
图 1-16 XMEDIA_IVE_BLOB_TYPE_U8 类型 xmedia_ive_blob_s (3 通道 2 帧示意图)	16
图 1-17 XMEDIA_IVE_BLOB_TYPE_YVU420SP 类型 xmedia_ive_blob_s (2 帧 YVU420SP 示意图)	17

图 1-18 XMEDIA_IVE_BLOB_TYPE_YVU422SP 类型 xmedia_ive_blob_s (2 帧 YVU422SP 示意图)	18
图 1-19 XMEDIA_IVE_BLOB_TYPE_VEC_S32 类型 xmedia_ive_blob_s (2 帧示意图)	18
图 1-20 XMEDIA_IVE_BLOB_TYPE_SEQ_S32 类型 xmedia_ive_blob_s(Num=N 帧示意图)	19
图 1-21 XMEDIA_IVE_BLOB_TYPE_U16 类型 xmedia_ive_blob_s(Num=N 帧示意图)	20
图 1-22 XMEDIA_IVE_BLOB_TYPE_S8 类型 xmedia_ive_blob_s(Num=N 帧示意图)	21
图 1-23 XMEDIA_IVE_BLOB_TYPE_F32 类型 xmedia_ive_blob_s(Num=N 帧示意图)	22
图 2-1 快速拷贝示意图	27
图 2-2 间隔拷贝示意图	28
图 2-3 filter 计算公式示意图	31
图 2-4 sobel 计算公式示意图	36
图 2-5 MagAndAng 计算示意图	40
图 2-6 dilate 计算公式示意图	44
图 2-7 erode 计算公式示意图	47
图 2-8 thresh 8 种阈值化模式示意图	51
图 2-9 thresh_S16 4 种阈值化模式示意图	63
图 2-10 thresh_U16 2 种阈值化模式示意图	65
图 2-11 16BitTo8Bit 4 种转换模式示意图	68
图 2-12 灰度图像 GMM 模型的内存配置示意图	84
图 2-13 RGB 图像 GMM 模型的内存配置示意图	85
图 2-14 灰度图像 gmm2 模型的内存配置示意图	88
图 2-15 RGB 图像 gmm2 模型的内存配置示意图	89
图 2-16 NormGrad 计算公式示意图	92
图 2-17 前景状态标志图形单个像素各比特位示意图	105
图 2-18 LBP 计算公式示意图	110
图 2-19 3 层金字塔 LK 光流计算示意图	115

表格目录

表 1-1 二维广义图像类型表 1	3
表 1-2 BLOB 内存排布类型表.....	13
表 4-1 智能加速引擎 API 错误码.....	250

1 概述

1.1 概述

IVE (Intelligent Video Engine) 是媒体处理芯片智能分析系统中的硬件加速模块。用户基于 IVE 开发智能分析方案可以加速智能分析，降低 CPU 占用。当前 IVE 提供的算子可以支撑开发视频诊断、周界防范等智能分析方案。

1.2 功能描述

1.2.1 重要概念

- 句柄(handle)
用户在调用算子创建任务时，系统会为每个任务分配一个 handle，用于标识不同的任务。
- 及时返回结果标志 b_instant
用户在创建某个任务后，希望及时得到该任务完成的信息，则需要在创建该任务时，将 b_instant 设置为 XMEDIA_TRUE。否则，如果用户不关心该任务是否完成，建议将 b_instant 设置为 XMEDIA_FALSE，这样可以与后续任务组链执行，减少中断次数，提升性能。
- 查询(query)
用户根据系统返回的 handle，调用 [xmedia_ive_query](#) 可以查询对应算子任务是否完成。
- 及时刷 cache

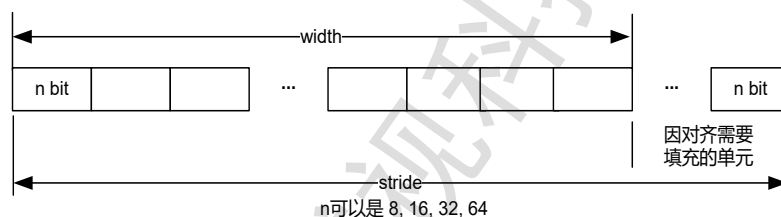
IVE 硬件只能从 DDR 中获取数据。如果用户在调用 IVE 任务时，访问空间可 cache 而且 CPU 曾经访问，为了保证 IVE 输入输出数据不被 CPU cache 干扰，此时用户需要调用 `xmedia_mmz_flush_cache`

接口刷 cache（详细信息请参见《MPP 媒体处理软件开发参考》），将数据从 cache 刷到 DDR，以供 IVE 使用。

- 跨度 (stride)：与图像或二维数据的 width 度量一致的量，如图 1-1 所示。
 - `xmedia_ive_image_s` 图像数据跨度，表示图像一行以“像素”计算的单元个数，“像素”位宽可以是 8bit，16bit 等。
 - `xmedia_ive_data_s` 二维数据跨度，表示二维数据一行的字节数，即为图 1-1 中 $n=8$ 的情况。

可以将 `xmedia_ive_data_s` 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。

须知

在使用 DDR4 时，为提高访存效率，建议首地址使用 256 字节对齐，stride 使用 256 “像素”的奇数倍对齐。如果是使用 64 位操作系统，需要使用到的 MMZ 地址必须是在一个 4GB 空间内，否则会出现异常。

- 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

`xmedia_ive_image_s`、`xmedia_ive_src_image_s`、`xmedia_ive_dst_image_s`，图像的类型参考 `xmedia_ive_image_type_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

`xmedia_ive_data_s`，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 `xmedia_ive_image_s` 可以转化为单个或多个 `xmedia_ive_data_s`。

- 一维数据

`xmedia_ive_mem_info_s`、`xmedia_ive_src_mem_info_s`、`xmedia_ive_dst_mem_info_s`，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

表1-1 二维广义图像类型表 1

类型	图像描述	内存地址	跨度
XMEDIA_IVE_IMAGE_TYPE_U8C1	8bit 无符号单通道图像，如图 1-2 所示	仅用到 <code>xmedia_ive_image_s</code> 中的 <code>au64PhyAddr[0]</code> 、 <code>au64VirAddr[0]</code>	仅用到 <code>u32Stride[0]</code>
XMEDIA_IVE_IMAGE_TYPE_S8C1	8bit 有符号单通道图像，如图 1-2 所示	仅用到 <code>xmedia_ive_image_s</code> 中的 <code>au64PhyAddr[0]</code> 、 <code>au64VirAddr[0]</code>	仅用到 <code>u32Stride[0]</code>
XMEDIA_IVE_IMAGE_TYPE_YUV420SP	YCbCr420 SemiPlannar 数据格式图像，如图 1-	内存地址仅用到 <code>xmedia_ive_image_s</code> 中的 <code>au64PhyAddr[0]</code> 、	跨度仅用到 <code>u32Stride[0]</code> (亮度跨度) 和

类型	图像描述	内存地址	跨度
	3 所示	au64VirAddr[0] (亮度 Y) 和 au64PhyAddr[1]、au64VirAddr[1] (色度, VU 间隔排列) ; 亮度和色度内存可以不连续, 但不推荐	u32Stride[1] (色度跨度)
XMEDIA_IVE_IMAGE_TYPE_YUV422SP	YcbCr422 SemiPlannar 数据格式图像, 如图 1-4 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0] (亮度 Y) 和 au64PhyAddr[1]、au64VirAddr[1] (色度, VU 间隔存储) ; 亮度和色度内存可以不连续, 但不推荐	跨度仅用到 u32Stride[0] (亮度跨度) 和 u32Stride[1] (色度跨度)
XMEDIA_IVE_IMAGE_TYPE_YUV420P	YCbCr420 Planar 数据格式图像, 如图 1-5 所示	内存地址用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0] (亮度 Y) , au64PhyAddr[1]、au64VirAddr[1] (色度 U) 和 au64PhyAddr[2]、au64VirAddr[2] (色度 V) ; Y、U、V 内存可不连续, 但不推荐	跨度用到 u32Stride[0] (亮度 Y 跨度)、u32Stride[1] (色度 U 跨度) 和 u32Stride[2] (色度 V 跨度)
XMEDIA_IVE_IMAGE_TYPE_YUV422P	YCbCr422 Planar 数据格式图像, 如图 1-6 所示	内存地址用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0] (亮度 Y) , au64PhyAddr[1]、au64VirAddr[1] (色度 U) 和 au64PhyAddr[2]、au64VirAddr[2] (色度 V)	跨度用到 u32Stride[0] (亮度跨度)、u32Stride[1] (色度 U 跨度) 和 u32Stride[2] (色度 V 跨度)

类型	图像描述	内存地址	跨度
XMEDIA_IVE_IMAGE_TYPE_S8C2_PACKAGE	8bit 有符号双通道且以 Package 格式存储的图像，如图 1-7 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]	跨度仅用到 u32Stride[0]
XMEDIA_IVE_IMAGE_TYPE_S8C2_PLANAR	8bit 有符号双通道且以 Planar 格式存储的图像，如图 1-8 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]和 au64PhyAddr[1]、au64VirAddr[1]	跨度仅用到 u32Stride[0]和 u32Stride[1]
XMEDIA_IVE_IMAGE_TYPE_S16C1	16bit 有符号单通道图像，如图 1-2 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]	跨度仅用到 u32Stride[0]
XMEDIA_IVE_IMAGE_TYPE_U16C1	16bit 无符号单通道图像，如图 1-2 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]	跨度仅用到 u32Stride[0]
XMEDIA_IVE_IMAGE_TYPE_U8C3_PACKAGE	8bit 无符号三通道且以 Package 格式存储的图像，如图 1-9 所示	内存地址仅用到 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]	跨度仅用到 u32Stride[0]
XMEDIA_IVE_IMAGE_TYPE_U8C3_PLANAR	8bit 无符号三通道且以 Planar 格式存储的图像，如图 1-10 所示；	内存地址用到了 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0]， au64PhyAddr[1]、au64VirAddr[1]和 au64PhyAddr[2]、au64VirAddr[2]；	跨度用到了 u32Stride[0]、u32Stride[1]和 u32Stride[2]；
IVE_IMAGE_TYPE_S32C1	32bit 有符号单通道图像，如图 1-2 所示；	内存地址仅用到了 xmedia_ive_image_s 中的 au64PhyAddr[0]、	跨度仅用到了 u32Stride[0]；

类型	图像描述	内存地址	跨度
		au64VirAddr[0];	
XMEDIA_IVE_IMAGE_TYPE_U32C1	32bit 无符号单通道图像, 如图 1-2 所示;	内存地址仅用到了 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0];	跨度仅用到了 u32Stride[0];
XMEDIA_IVE_IMAGE_TYPE_S64C1	64bit 有符号单通道图像, 如图 1-2 所示;	内存地址仅用到了 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0];	跨度仅用到了 u32Stride[0];
XMEDIA_IVE_IMAGE_TYPE_U64C1	64bit 无符号单通道图像, 如图 1-2 所示;	内存地址仅用到了 xmedia_ive_image_s 中的 au64PhyAddr[0]、au64VirAddr[0];	跨度仅用到了 u32Stride[0];

- 特殊输出数据类型
 - Integ 组合输出 (IVE_INTEG_OUT_CTRL_COMBINE)
 - 用 XMEDIA_IVE_IMAGE_TYPE_U64C1 类型的 [xmedia_ive_image_s](#), S (图像和) 占低 28bit, SQ (图像平方和) 占高 36bit。格式如图 1-13 所示。
 - 直方图输出如图 1-14 所示。

图1-2 XMEDIA_IVE_IMAGE_TYPE_U8C1\XMEDIA_IVE_IMAGE_TYPE_S8C1\XMEDIA_IVE_IMAGE_TYPE_S16C1\XMEDIA_IVE_IMAGE_TYPE_U16C1\XMEDIA_IVE_IMAGE_TYPE_S32C1\XMEDIA_IVE_IMAGE_TYPE_U32C1\XMEDIA_IVE_IMAGE_TYPE_S64C1\XMEDIA_IVE_IMAGE_TYPE_U64C1 类型的 [xmedia_ive_image_s](#) 图像

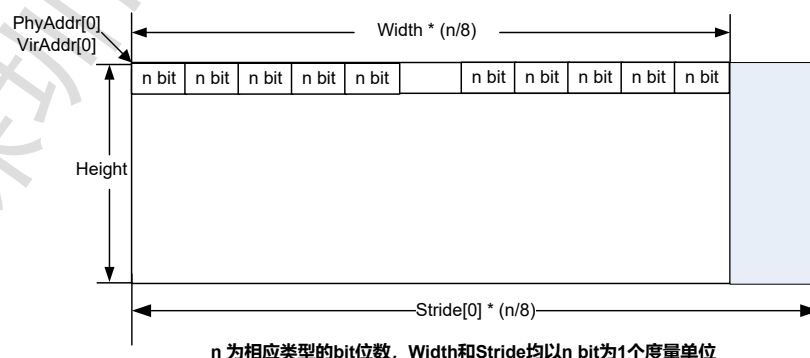
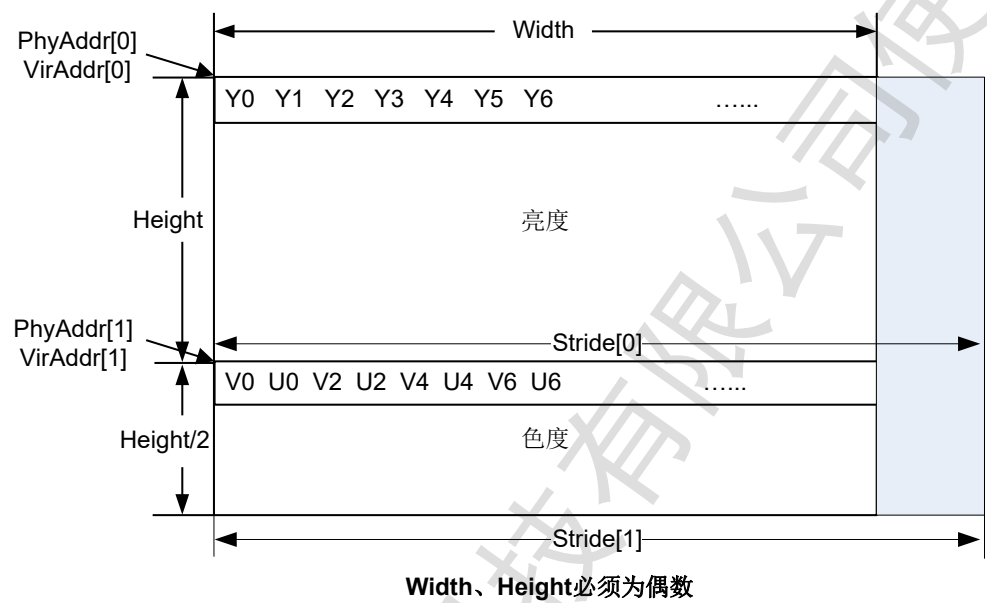
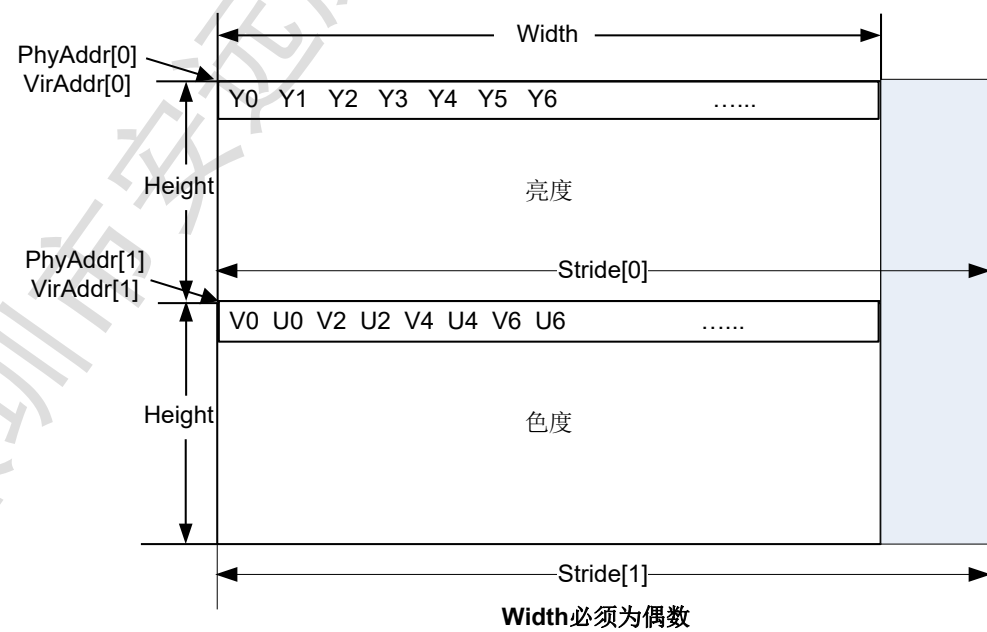


图1-3 XMEDIA_IVE_IMAGE_TYPE_YUV420SP 类型的 xmedia_ive_image_s 图像



注：格式与 xmedia_video_pixel_format 的 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_420 对应，对于其他芯片，格式与 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_420 对应。

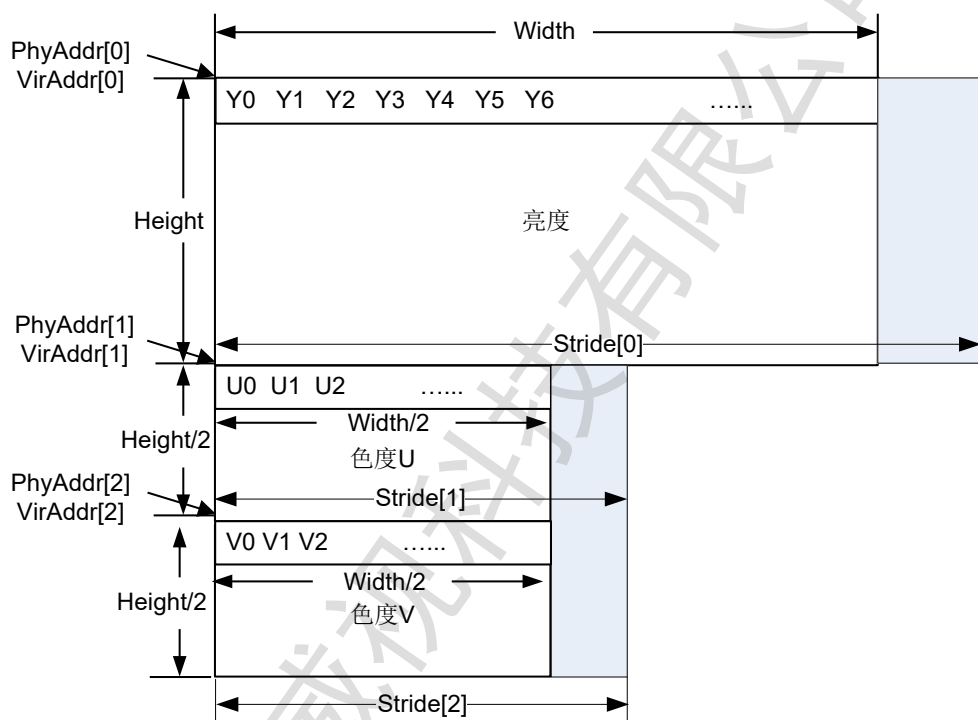
图1-4 XMEDIA_IVE_IMAGE_TYPE_YUV422SP 类型的 xmedia_ive_image_s 图像



注：

- 这里 V 在前, U 在后, PhyAddr[2]和 VirAddr[2]可配置为 U 的首地址, 即 PhyAddr[1]+1 和 VirAddr[1]+1。
- 格式与 xmedia_video_pixel_format 的 XMEDIA_VIDEO_PIXEL_FMT_YVU_SEMIPLANAR_422 对应, 对于其他芯片, 格式与 XMEDIA_VIDEO_PIXEL_FMT_YUV_SEMIPLANAR_422 对应。

图1-5 XMEDIA_IVE_IMAGE_TYPE_YUV420P 类型的 xmedia_ive_image_s 图像



Width、Height必须为偶数

图1-6 XMEDIA_IVE_IMAGE_TYPE_YUV422P 类型的 xmedia_ive_image_s 图像

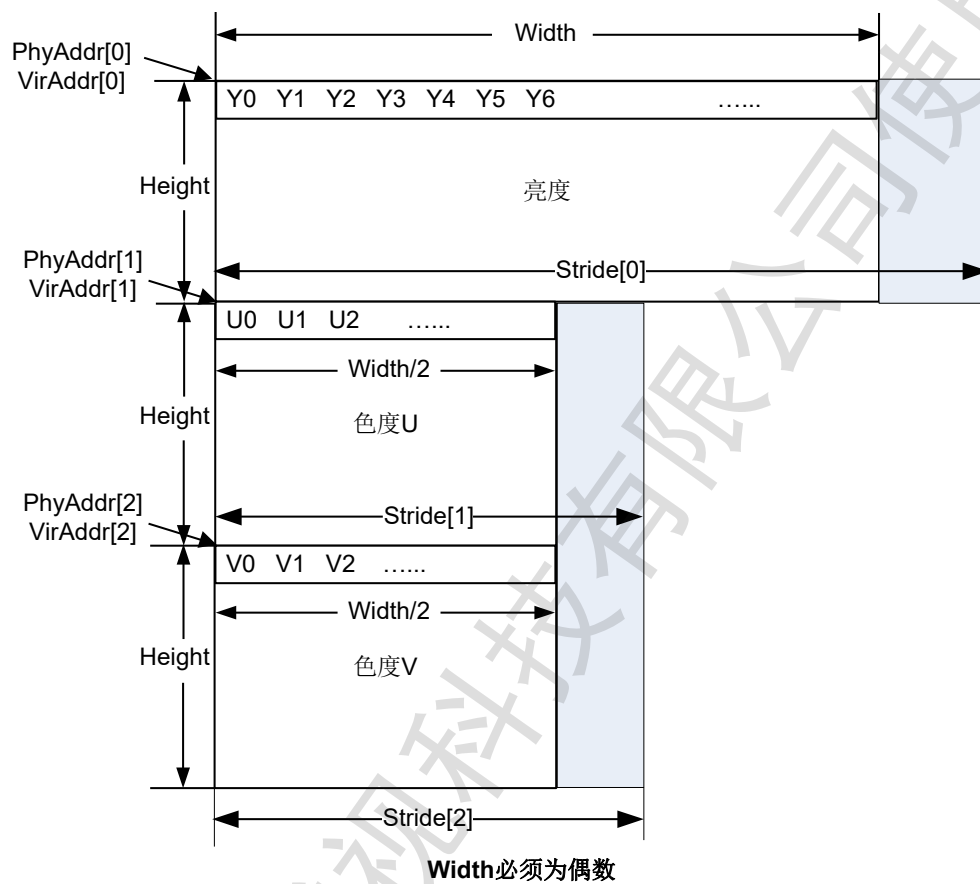


图1-7 XMEDIA_IVE_IMAGE_TYPE_S8C2_PACKAGE 类型的 xmedia_ive_image_s 图像

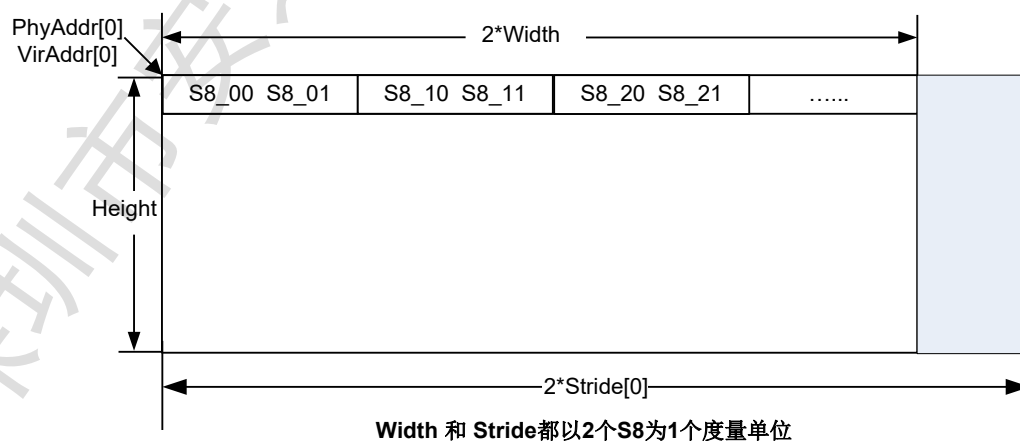


图1-8 XMEDIA_IVE_IMAGE_TYPE_S8C2_PLANAR 类型的 xmedia_ive_image_s 图像

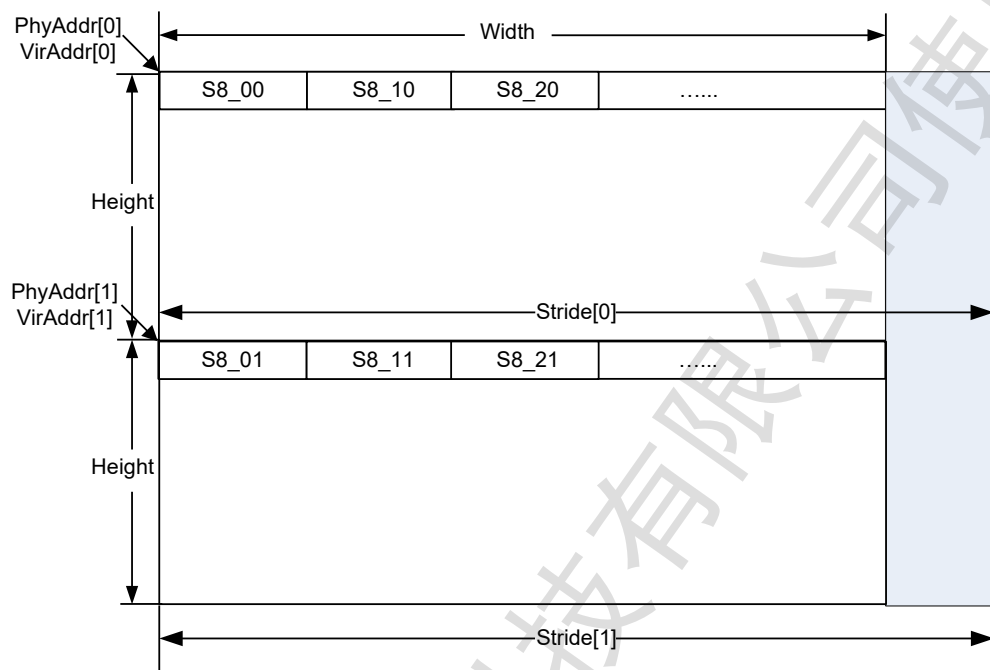
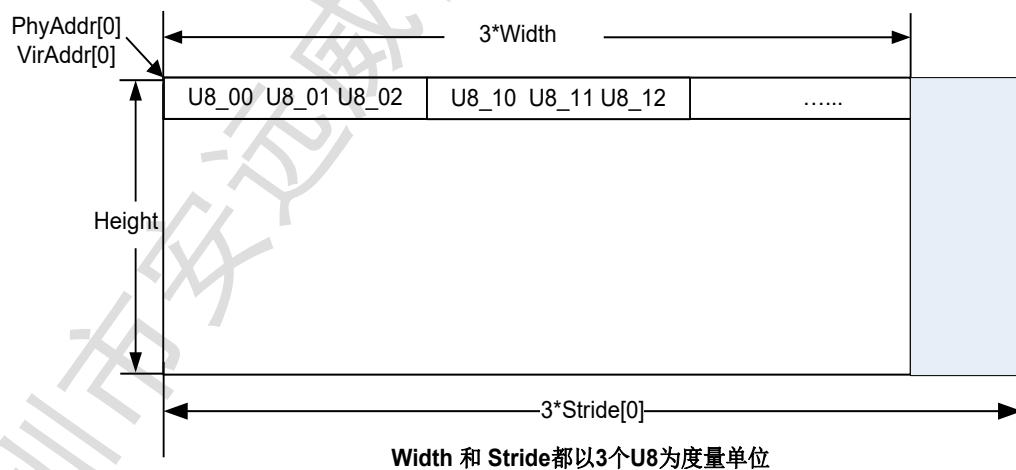
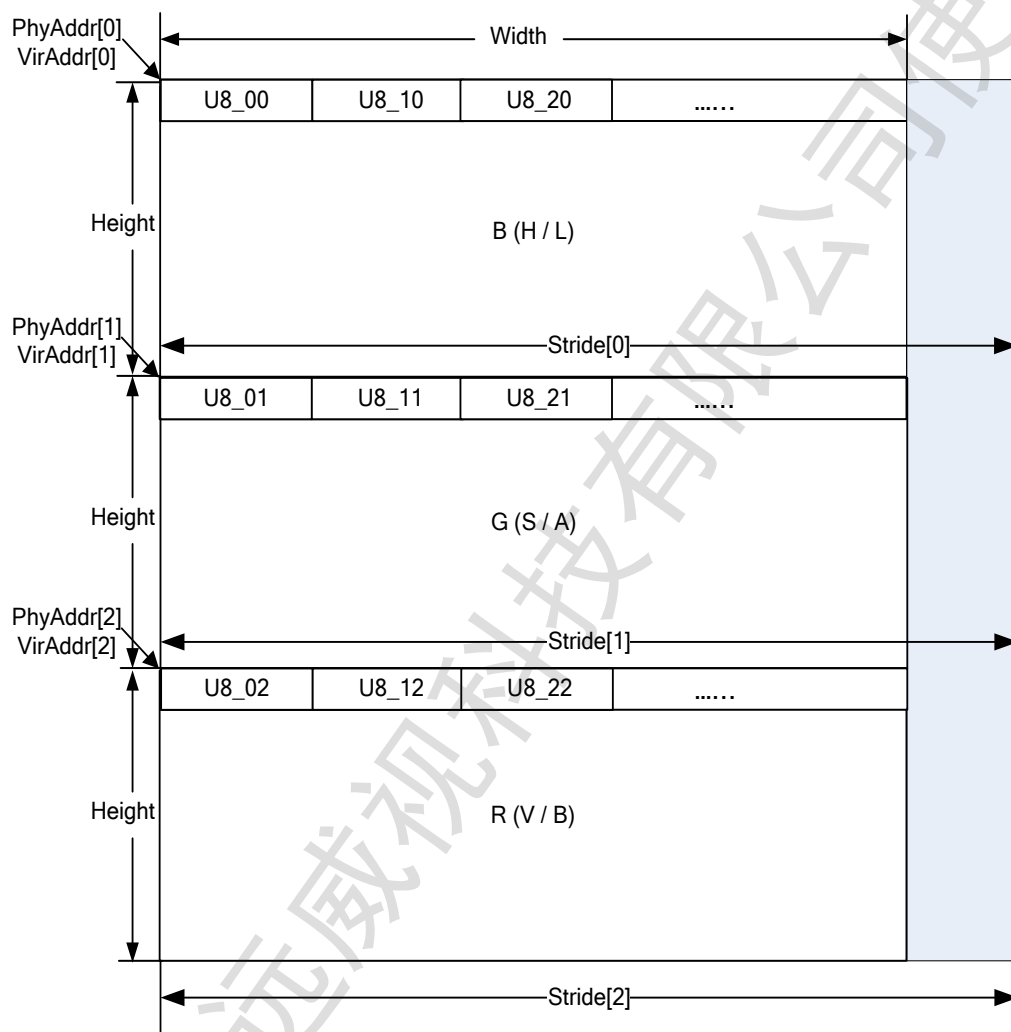


图1-9 XMEDIA_IVE_IMAGE_TYPE_U8C3_PACKAGE 类型的 xmedia_ive_image_s 图像



注：对于 RGB_PACKAGE 图像，是以 “B0G0R0B1G1R1...” 形式存储，B 在最前面；
 对于 HSV_PACKAGE 图像，是以 “H0S0V0H1S1V1...” 形式存储，H 在最前面；
 对于 LAB_PACKAGE 图像，是以 “L0A0B0L1A1B1...” 形式存储，L 在最前面。

图1-10 XMEDIA_IVE_IMAGE_TYPE_U8C3_PLANAR 类型的 xmedia_ive_image_s 图像



注：对于 RGB_PLANAR 图像，指针数组 `VirAddr[3]`按顺序分别存储 B、G、R 的指针，而数组 `Stride[3]`分别为 B、G、R 的跨度；

对于 HSV_PLANAR 图像，指针数组 `VirAddr[3]`按顺序分别存储 H、S、V 的指针，而数组 `Stride[3]`分别为 H、S、V 的跨度；

对于 LAB_PLANAR 图像，指针数组 `VirAddr[3]`按顺序分别存储 L、A、B 的指针，而数组 `Stride[3]`分别为 L、A、B 的跨度；

图1-11 xmedia_ive_data_s 类型的数据内存示意

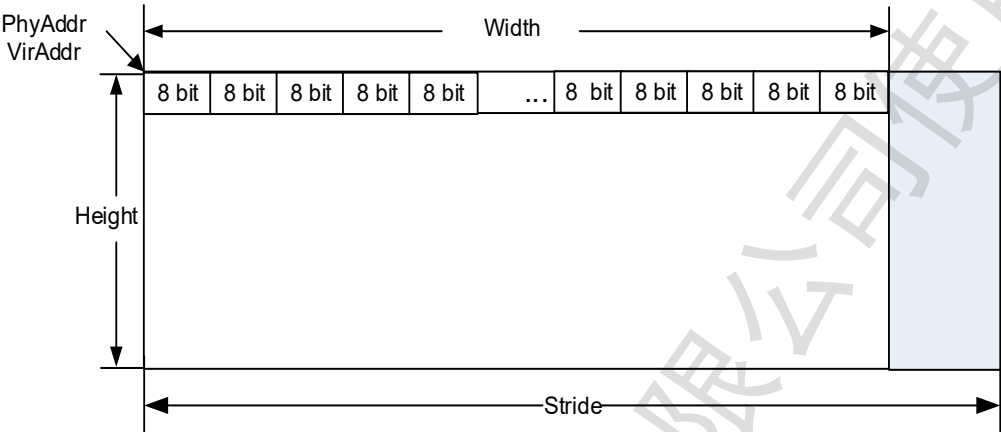


图1-12 xmedia_ive_mem_s 类型的数据内存示意

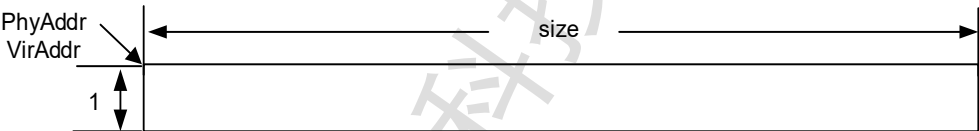


图1-13 积分图 (XMEDIA_IVE_IMAGE_TYPE_U64C1) 组合输出示意

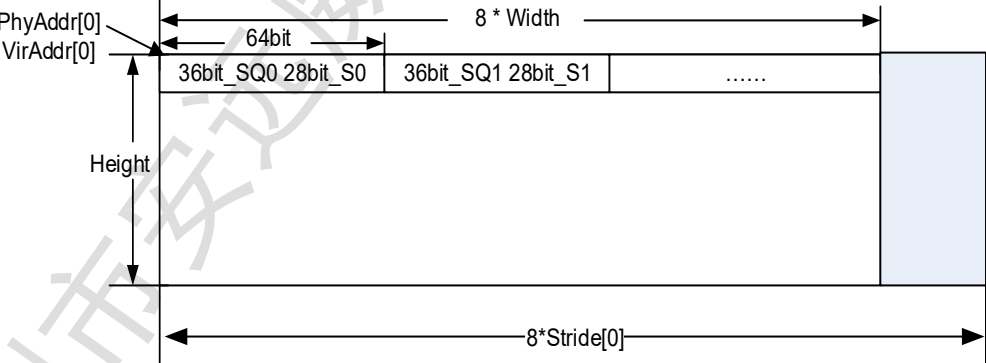
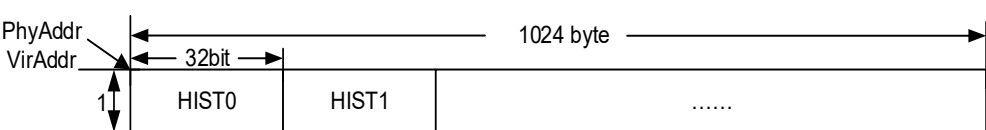


图1-14 直方图输出格式示意



- 块数据类型

[xmedia_ive_blob_s](#)、[xmedia_ive_src_blob_s](#)、[xmedia_ive_dst_blob_s](#)，类型参考 [xmedia_ive_blob_type_e](#)，具体的内存分配如图 1-15 ~ 图 1-23 所示。

• BLOB 内存排布类型

表1-2 BLOB 内存排布类型表

类型	BLOB 描述
XMEDIA_IVE_BLOB_TYPE_S32	多帧有符号 32bit 多通道数据 Planar 格式存储顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn 表示单帧图像通道数，如图 1-15 所示。
XMEDIA_IVE_BLOB_TYPE_U8	多帧无符号 8bit 多通道数据 Planar 格式存储顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn 表示单帧图像通道数，如图 1-16 所示。
XMEDIA_IVE_BLOB_TYPE_YVU420SP	多帧 YCbCr420 SemiPlannar 数据格式图像顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn=3，如图 1-17 所示。色度部分 V 在前，U 在后。注意此时高、宽必须为偶数。
XMEDIA_IVE_BLOB_TYPE_YVU422SP	多帧 YCbCr422 SemiPlannar 数据格式图像顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn=3，如图 1-18 所示。色度部分 V 在前，U 在后。注意此时宽必须为偶数。
XMEDIA_XMEDIA_IVE_BLOB_TYPE_VEC_S32	多帧有符号 32bit 向量数据顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示向量数据维度，

类型	BLOB 描述
	u32height 表示单帧有多少个向量（一般 u32height=1），u32chn=1，如图 1-19 所示。
XMEDIA_IVE_BLOB_TYPE_SEQ_S32	多段有符号 32bit 序列数据排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示段数，u32dim 表示序列向量数据维度，u64viraddrstep 是一个 u32_num 长度的数组地址，数组元素表示每段序列有多少个向量，如图 1-20 所示。
XMEDIA_IVE_BLOB_TYPE_U16	多帧无符号 16bit 多通道数据 Planar 格式存储顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn 表示单帧图像通道数，如图 1-21 所示。
XMEDIA_IVE_BLOB_TYPE_S8	多帧有符号 8bit 多通道数据 Planar 格式存储顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn 表示单帧图像通道数，如图 1-22 所示。
XMEDIA_IVE_BLOB_TYPE_F32	多帧浮点 32bit 多通道数据 Planar 格式存储顺序排布。此时 xmedia_ive_blob_s 结构体中，u32_num 表示帧数，u32width 表示图像宽，u32height 表示图像高，u32chn 表示单帧图像通道数，如图 1-23 所示。

图1-15 XMEDIA_IVE_BLOB_TYPE_S32 类型 xmedia_ive_blob_s (2 通道 2 帧示意图)

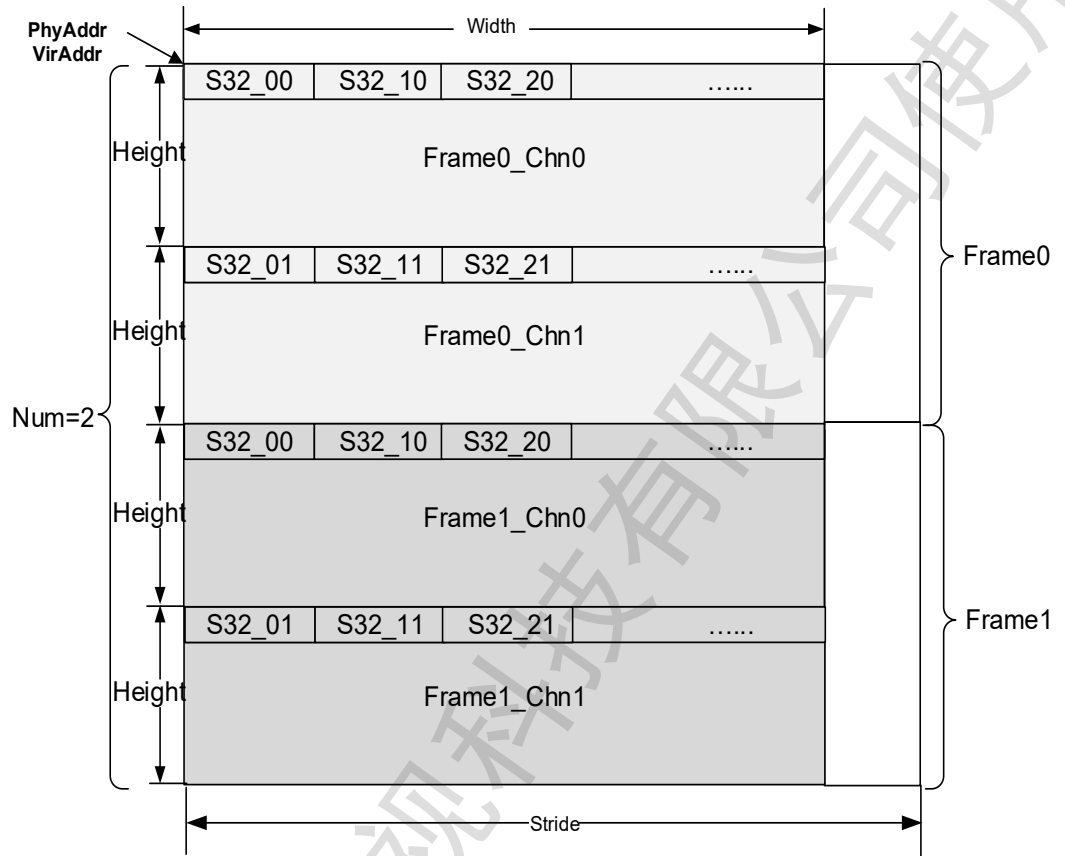
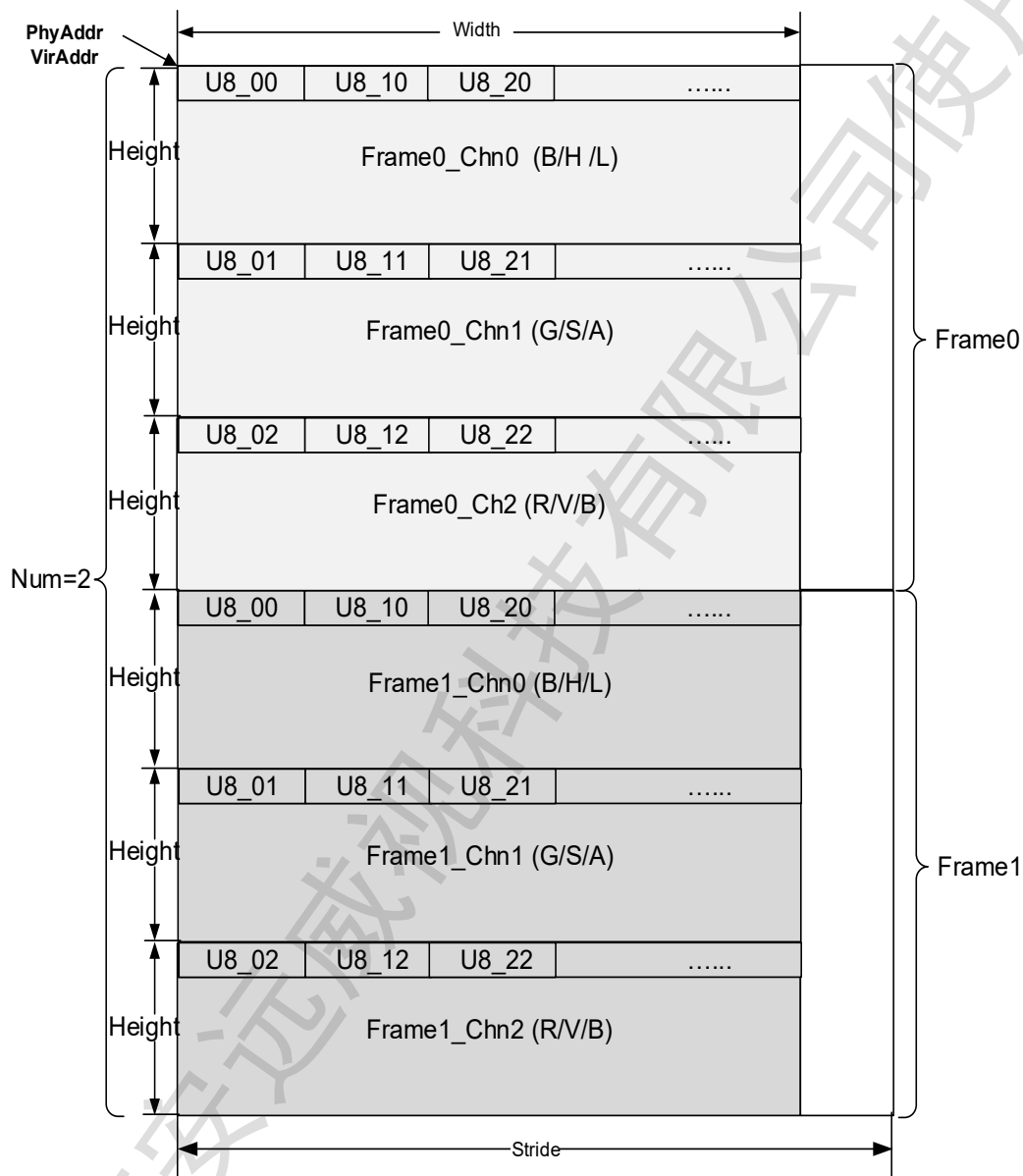


图1-16 XMEDIA_IVE_BLOB_TYPE_U8 类型 xmedia_ive_blob_s (3 通道 2 帧示意图)



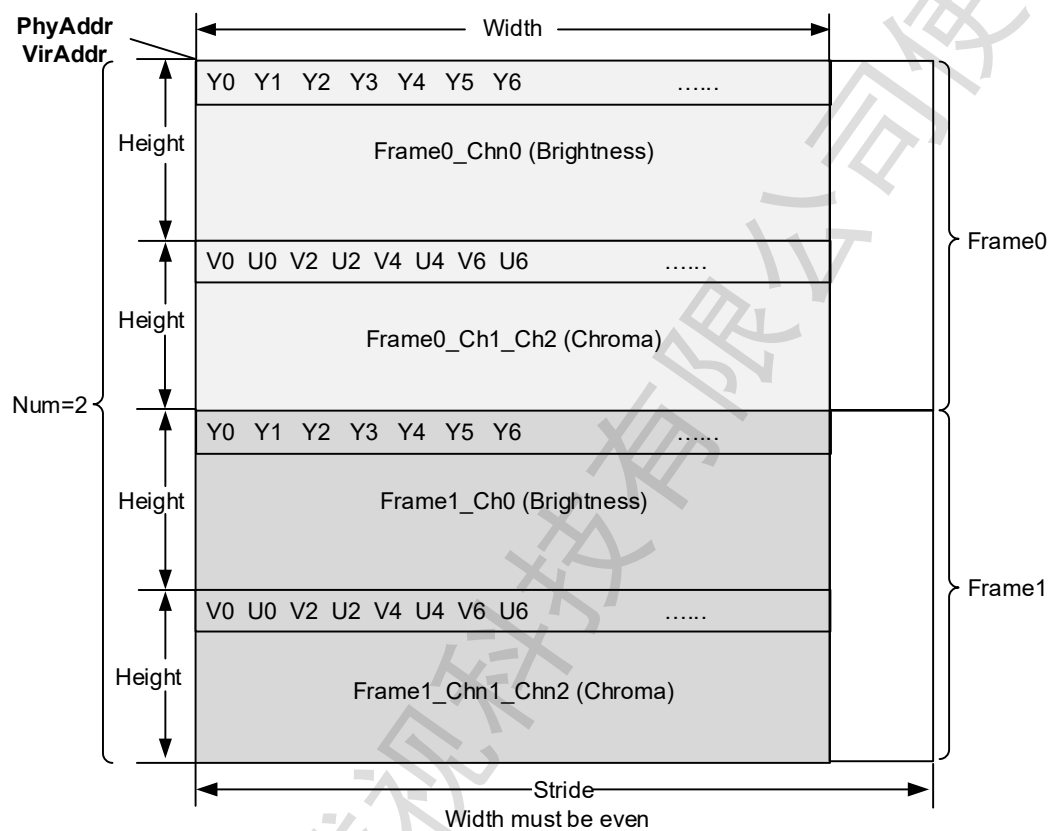
注：典型的 RGB\HSV\LAB 图像 Planar 格式存储，IVE 默认以图示的 B\G\R、H\S\V、L\A\B 顺序存储。

图1-17 XMEDIA_IVE_BLOB_TYPE_YVU420SP 类型 xmedia_ive_blob_s (2 帧 YVU420SP 示意图)



注：这里 V 在前，U 在后。

图1-18 XMEDIA_IVE_BLOB_TYPE_YVU422SP 类型 xmedia_ive_blob_s (2 帧 YVU422SP 示意图)



注：这里 V 在前，U 在后。

图1-19 XMEDIA_IVE_BLOB_TYPE_VEC_S32 类型 xmedia_ive_blob_s (2 帧示意图)

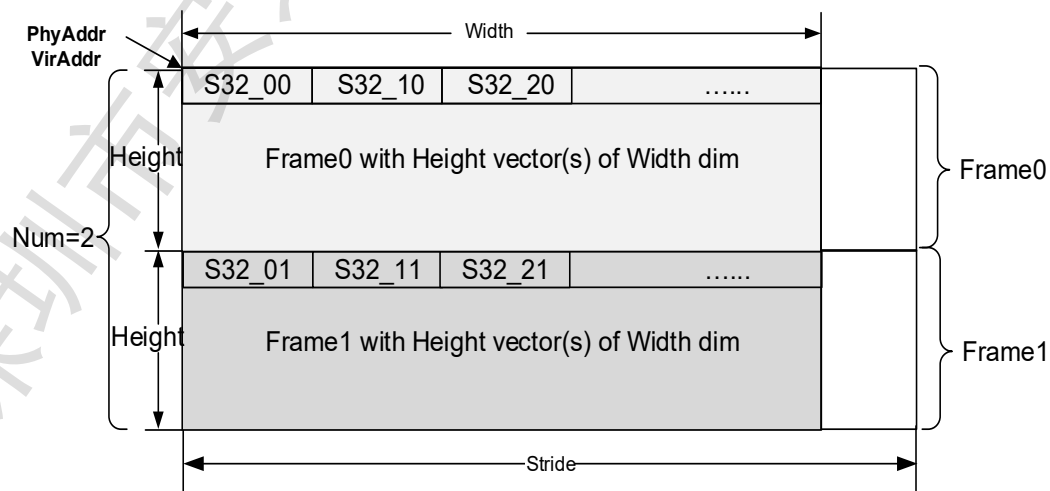


图1-20 XMEDIA_IVE_BLOB_TYPE_SEQ_S32 类型 xmedia_ive_blob_s(Num=N 帧示意图)

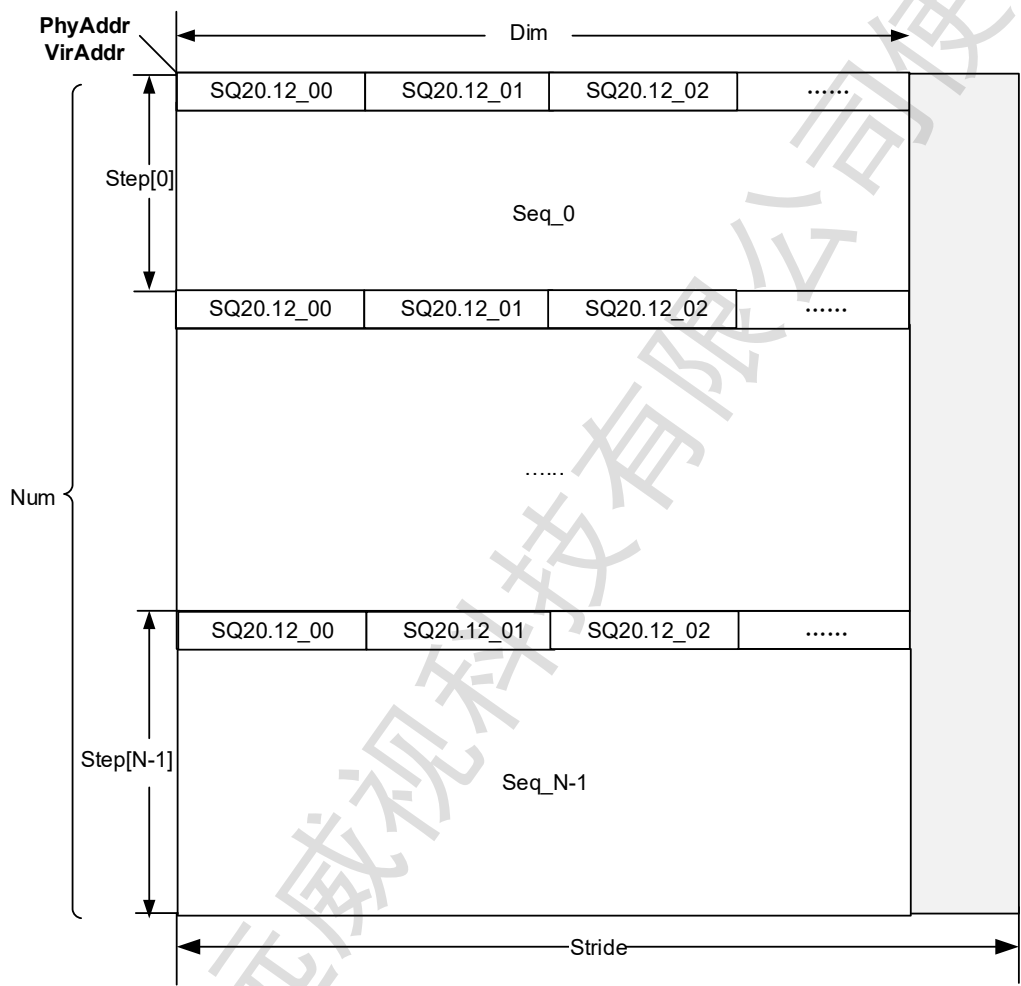


图1-21 XMEDIA_IVE_BLOB_TYPE_U16 类型 xmedia_ive_blob_s(Num=N 帧示意图)

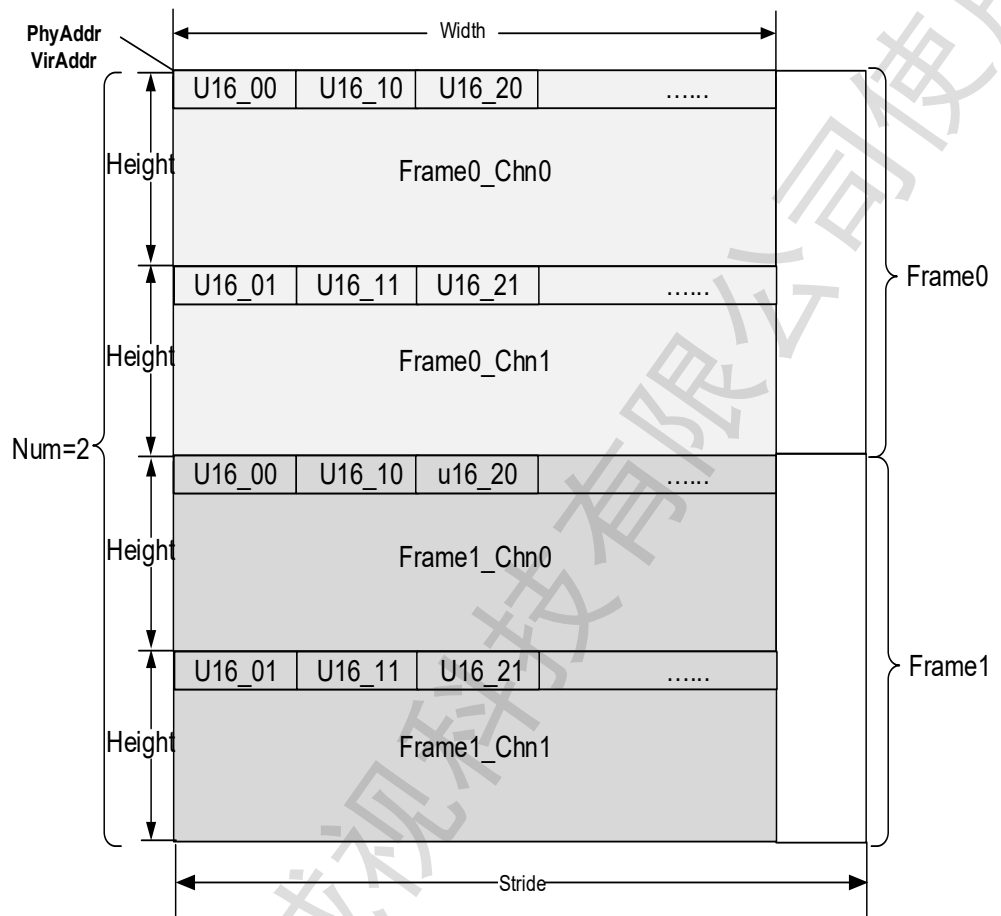


图1-22 XMEDIA_IVE_BLOB_TYPE_S8 类型 xmedia_ive_blob_s(Num=N 帧示意图)

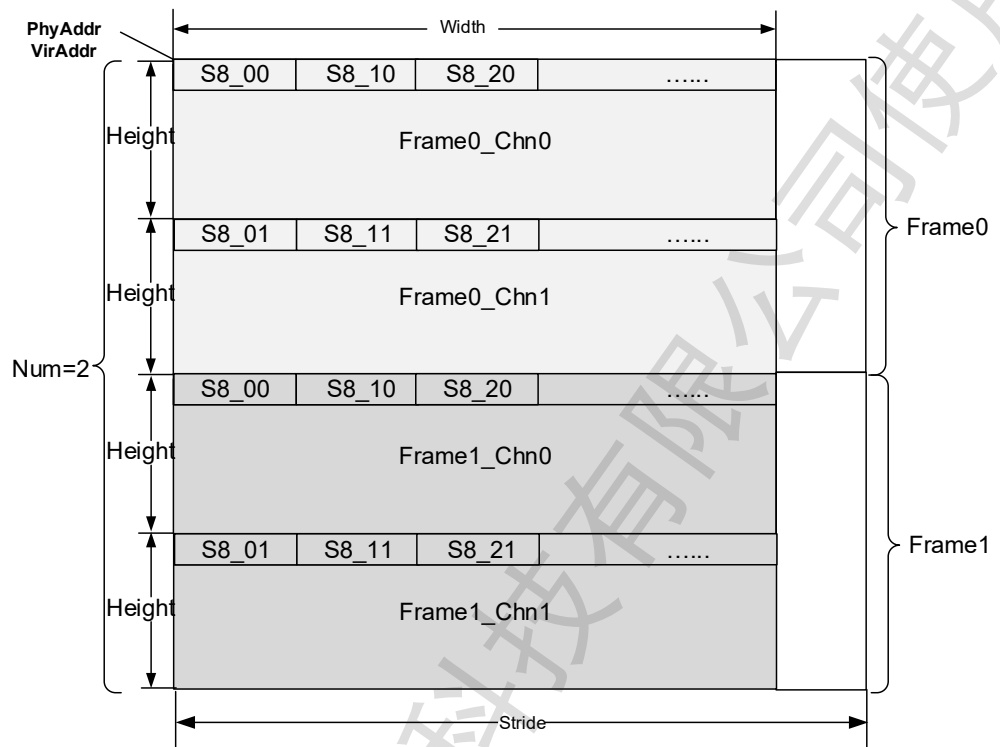
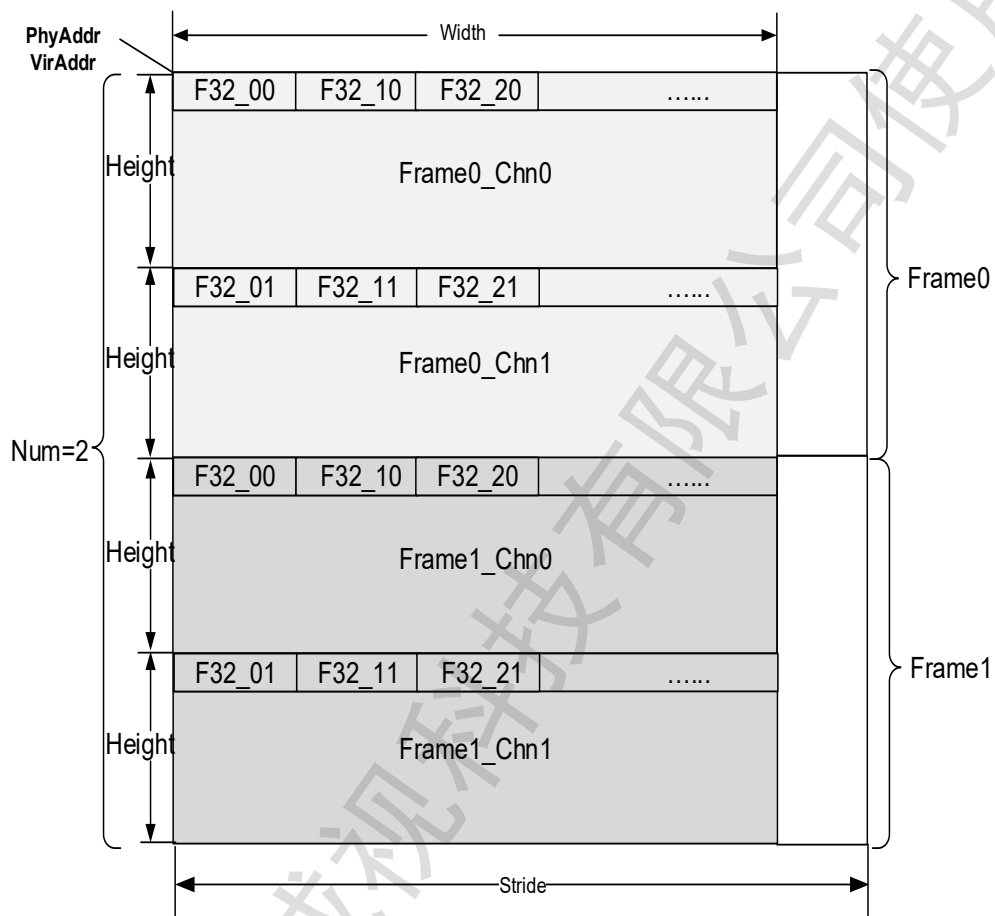


图1-23 XMEDIA_IVE_BLOB_TYPE_F32 类型 xmedia_ive_blob_s(Num=N 帧示意图)



1.2.2 使用示意

- 用户根据需求调用相应的算子接口创建任务，指定 b_instant 类型，并记录该任务返回的 handle 号。
- 根据返回的 handle 号，指定阻塞方式，可以查询到该任务的完成状态。
具体可参见 [xmedia_ive_query](#) 中的【举例】。

2 API 参考

IVE 模块提供了创建任务和查询任务的基本接口。

该功能模块提供以下 API:

- `xmedia_ive_dma`: 创建直接内存访问任务。
- `xmedia_ive_filter`: 创建 5x5 模板滤波任务。
- `xmedia_ive_csc`: 创建色彩空间转换任务。
- `xmedia_ive_sobel`: 创建 5x5 模板 sobel-like 梯度计算任务。
- `xmedia_ive_magandang`: 创建 5x5 模板计算梯度幅值与幅角任务。
- `xmedia_ive_dilate`: 创建膨胀任务。
- `xmedia_ive_erode`: 创建腐蚀任务。
- `xmedia_ive_thresh`: 创建图像二值化任务。
- `xmedia_ive_and`: 创建两图像相与任务。
- `xmedia_ive_sub`: 创建两图像相减任务。
- `xmedia_ive_or`: 创建两图像相或任务。
- `xmedia_ive_integ`: 创建积分图统计任务。
- `xmedia_ive_hist`: 创建直方图统计任务。
- `xmedia_ive_thresh_s16`: 创建 S16 数据到 8bit 数据阈值化任务。
- `xmedia_ive_thresh_u16`: 创建 U16 数据到 U8 数据阈值化任务。
- `xmedia_ive_16bit_to_8bit`: 创建 16bit 数据到 8bit 数据线性转化任务。
- `xmedia_ive_ord_stat_filter`: 创建 3x3 模板顺序统计量滤波任务。
- `xmedia_ive_map`: 创建 Map (映射 U8->U8\U8->U16\U8->S16 赋值) 任务。
- `xmedia_ive_equalize_hist`: 创建灰度图像的直方图均衡化计算任务。

- `xmedia_ive_add`: 创建两灰度图像的加权加计算任务。
- `xmedia_ive_xor`: 创建两二值图的异或计算任务。
- `xmedia_ive_ncc`: 创建两相同分辨率图像的归一化互相关系数计算任务。
- `xmedia_ive_ccl`: 创建二值图像的连通区域标记任务。
- `xmedia_ive_gmm`: 创建 GMM 背景建模任务。
- `xmedia_ive_gmm2`: 创建 gmm2 背景建模任务。
- `xmedia_ive_normgrad`: 创建归一化梯度计算任务，梯度均分量均归一化到 S8。
- `xmedia_ive_st_candi_corner`: 灰度图像 Shi-Tomasi-like 角点计算的前半部：计算候选角点。
- `xmedia_ive_st_corner`: 灰度图像 Shi-Tomasi-like 角点计算的后半部：按规则挑选角点。
- `xmedia_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。
- `xmedia_ive_resize_csc_letterbox`: 创建图像缩放、色域转换、加边框复合任务，支持双线性缩放，支持 yuv2rgb 色彩空间转换。
- `xmedia_ive_grad_fg`: 根据背景图像和当前帧图像的梯度信息计算梯度前景图像。
- `xmedia_ive_match_bg_model`: 基于 CodeBook 演进的背景模型匹配。
- `xmedia_ive_update_bg_model`: 基于 CodeBook 演进的背景模型更新。
- `xmedia_ive_lbp`: 创建计算图像的 LBP 特征任务。
- `xmedia_ive_acc`: 创建两图像像素累加运算任务。
- `xmedia_ive_lk_optical_flow_pyr`: 创建 LK 光流跟踪计算任务
- `xmedia_ive_and_ccl`: 创建两二值图像相与&连通区域标记组合计算任务
- `xmedia_ive_or_ccl`: 创建两二值图像相或&连通区域标记组合计算任务
- `xmedia_ive_xor_ccl`: 创建两二值图的异或&连通区域标记组合计算任务。
- `xmedia_ive_sub_ccl`: 创建两灰度图像相减&连通区域标记组合计算任务。
- `xmedia_ive_add_ccl`: 创建两二值图像相与&连通区域标记组合计算任务。
- `xmedia_ive_sub_hist`: 创建两灰度图像相减&直方图统计组合计算任务。
- `xmedia_ive_sub_thresh`: 创建两灰度图像相减&阈值化组合计算任务。
- `xmedia_ive_sub_integ`: 创建两灰度图像相减&积分图计算组合计算任务。
- `xmedia_ive_sub_map`: 创建两灰度图像相减&map（映射赋值）组合计算任务。

- [xmedia_ive_add_hist](#): 创建两二值图像相与&直方图统计组合计算任务。
- [xmedia_ive_add_thresh](#): 创建两二值图像相与&阈值化组合计算任务。
- [xmedia_ive_add_integ](#): 创建两二值图像相与&积分图计算组合计算任务。
- [xmedia_ive_add_map](#): 创建两二值图像相与&map（映射赋值）组合计算任务。
- [xmedia_ive_filter_thresh](#): 创建 5x5 模板滤波 & 阈值化组合计算任务。
- [xmedia_ive_ord_stat_filter_thresh](#): 创建 3x3 模板顺序统计量滤波 & 阈值化组合任务计算任务。
- [xmedia_ive_sobel_thresh_s16](#): 创建 5x5 模板 sobel-like 梯度计算 & s16 数据到 8bit 数据的阈值化组合计算任务。
- [xmedia_ive_mag_and_ang_thresh_u16](#): 创建 5x5 模板梯度幅值& 幅角计算 & u16 数据到 u8 数据的阈值化组合计算任务。
- [xmedia_ive_mag_and_ang_16bit_to_8bit](#): 创建 5x5 模板梯度幅值& 幅角计算 & 16bit 图像数据到 8bit 图像数据的线性转化组合计算任务。
- [xmedia_ive_thresh_dilate](#): 创建灰度图像阈值化& 5x5 模板膨胀组合计算任务。
- [xmedia_ive_thresh_erode](#): 创建灰度图像阈值化& 5x5 模板腐蚀组合计算任务。
- [xmedia_ive_query](#): 查询已创建任务完成情况。

xmedia_ive_dma

【描述】

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

【语法】

```
xmedia_s32 xmedia_ive_dma(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_data_s
*pst_src, xmedia_ive_dst_data_s *pst_dst, xmedia_ive_dma_ctrl_s *pst_dma_ctrl, xmedia_bool
b_instant);
```

【参数】

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

参数名称	描述	输入/输出
pst_src	源数据指针。 不能为空。	输入(set 模式下同时 也是输出)
pst_dst	输出数据指针。 copy 模式下不能为空。	输出
pst_dma_ctrl	DMA 控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

注:

- copy 模式是指 XMEDIA_IVE_DMA_MODE_DIRECT_COPY 和 XMEDIA_IVE_DMA_MODE_INTERVAL_COPY 模式;
- set 模式是指 XMEDIA_IVE_DMA_MODE_SET_3BYTE 和 XMEDIA_IVE_DMA_MODE_SET_8BYTE 模式。

参数名称	支持类型	地址对齐	分辨率
pst_src	xmedia_ive_data_s	1 byte	见【芯片差异】
pst_dst	xmedia_ive_dst_data_s	1 byte	直接拷贝时同 pst_src; 间隔拷贝时比 pst_src 小。

【返回值】

返回值	描述
0	成功。
非 0	失败, 参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	pst_src 的分辨率在 1x1 ~ 1920x1080
XM7206V11A	pst_src 的分辨率在 1x1 ~ 4096x4096

【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.a

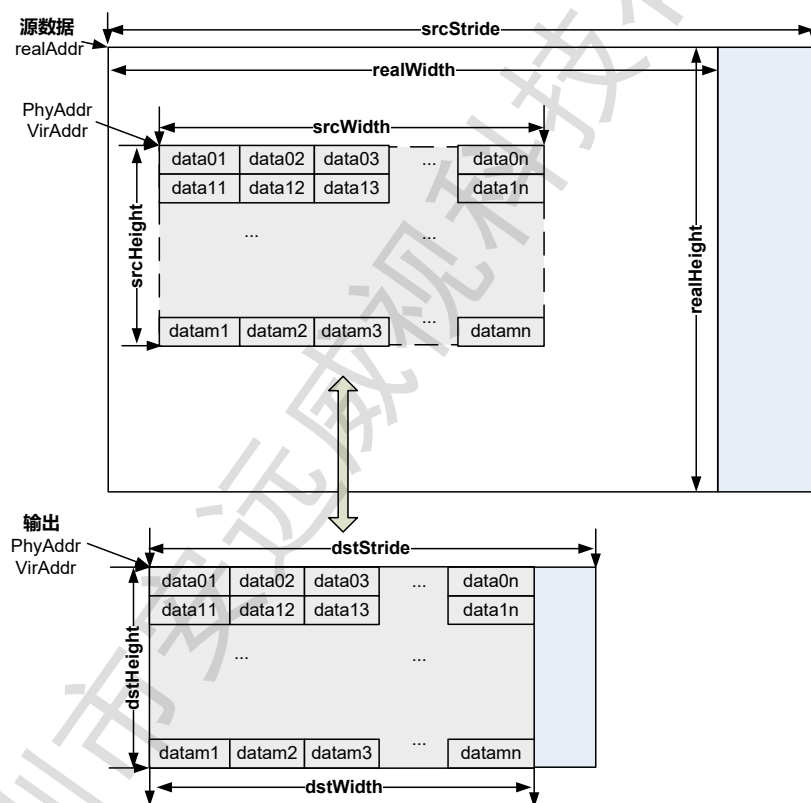
【注意】

- XMEDIA_IVE_DMA_MODE_DIRECT_COPY: 快速拷贝模式
可实现从大块内存中扣取小块内存, 如图 2-1 所示, 计算公式如下:

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

其中 $I(x, y)$ 对应 pst_src , $I_{out}(x, y)$ 对应 pst_dst 。

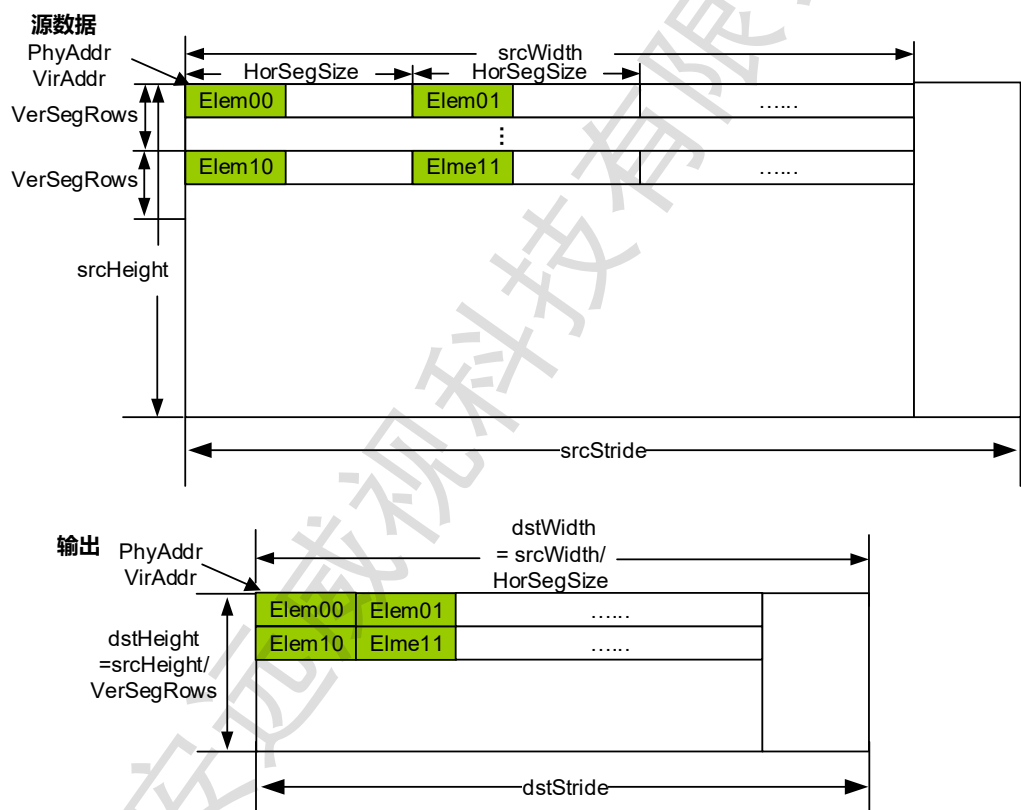
图2-1 快速拷贝示意图



- XMEDIA_IVE_DMA_MODE_INTERVAL_COPY: 间隔拷贝模式
 - 要求源数据宽度为 u8_hor_seg_size 的倍数;
 - 间隔拷贝的方式: 将每 u16_ver_seg_rows 行中第一行数据分割为 u8_hor_seg_size 大小的段, 拷贝每段中的前 u8_elem_size 大小的字节。如图 2-2 所示。

- XMEDIA_IVE_DMA_MODE_SET_3BYTE: 3 字节填充模式
 仅使用 pst_src, 用 u64_val 的低 3 字节对源数据进行填充操作; 当一行末尾不够 3 字节时, 用 u64_val 的低字节填充。
- XMEDIA_IVE_DMA_MODE_SET_8BYTE: 8 字节填充模式
 仅使用 pst_src, 用 u64_val 对源数据进行填充操作; 当一行的末尾不足 8 字节时, 用 u64_val 的低字节填充。

图2-2 间隔拷贝示意图



【举例】

无。

【相关主题】

无。

xmedia_ive_filter

【描述】

创建 5x5 模板滤波任务，通过配置不同的模板系数，可以实现不同的滤波。

【语法】

```
xmedia_s32 xmedia_ive_filter(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_filter_ctrl_s  
*pstflt_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pstflt_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1、YUV420SP、YUV422SP	见【芯片差异】	32x32 ~ 1920x1024
pst_dst	同 pst_src	见【芯片差异】	同 pst_src

注：U8C1\YUV420SP\YUV422SP 均为 [xmedia_ive_image_type_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。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

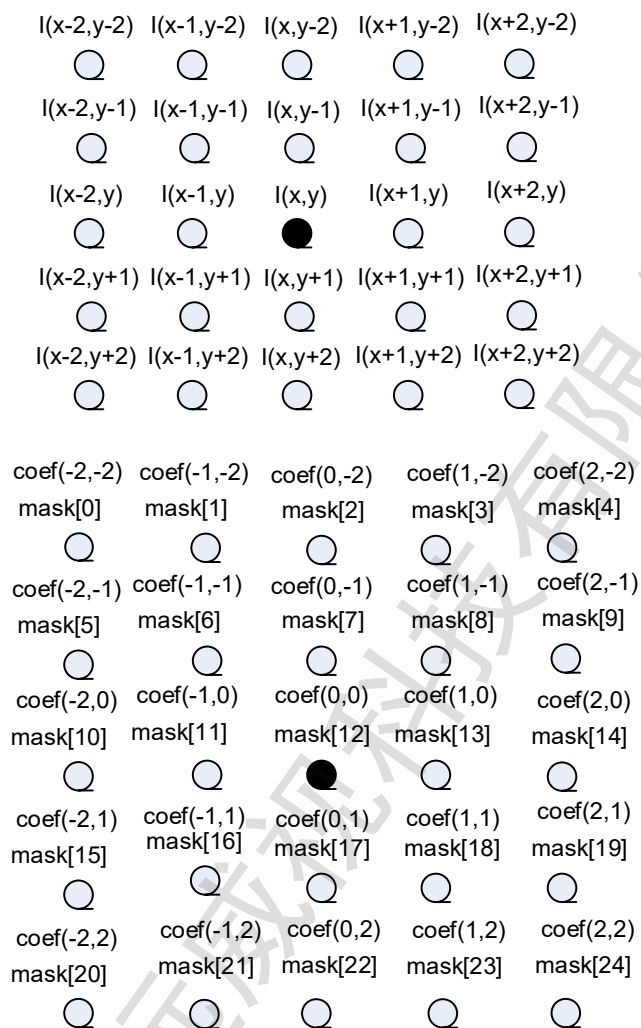
【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.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



$$I_{out}(x,y) = \left\{ \sum_{-2 < j < 2} \sum_{-2 < i < 2} I(x+i, y+j) \cdot coef(i,j) \right\} \gg \text{norm}$$

其中, $I(x,y)$ 对应 pst_src , $I_{out}(x,y)$ 对应 pst_dst , $coef(mask)$ 对应 $pstFltCtrl$ 中的 $as8_mask[25]$, norm 对应 $pstFltCtrl$ 中的 $u8_norm$ 。

- 经典高斯模板如下。

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

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

【举例】

无。

【相关主题】

[xmedia_ive_ord_stat_filter](#)

xmedia_ive_csc

【描述】

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

【语法】

```
xmedia_s32 xmedia_ive_csc(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s
*pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_csc_ctrl_s *pst_csc_ctrl, xmedia_bool
b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_csc_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	YUV420SP、YUV422SP、U8C3_PLANAR、U8C3_PACKAGE	见【芯片差异】	4x4 ~ 1920x1080
pst_dst	U8C3_PLANAR、U8C3_PACKAGE、YUV420SP、YUV422SP	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐，只支持 YUV2RGB \RGB2YUV，且限制 RGB 为 U8C3_PACKAGE。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.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 [xmedia_ive_csc_mode_e](#)。
- YUV2HSV、YUV2LAB 参考 OpenCV 中的实现方法。



说明

本文档中所提到的 OpenCV，均指 OpenCV 2.4.8 版本。

【举例】

无。

【相关主题】

无

xmedia_ive_sobel

【描述】

创建 5x5 模板 sobel-like 梯度计算任务。

【语法】

```
xmedia_s32 xmedia_ive_sobel(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst_h, xmedia_ive_dst_image_s  
*pst_dst_v, xmedia_ive_sobel_ctrl_s *pst_sobel_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst_h	由模板直接滤波得到的梯度分量图像 H 指针。 根据 pstSobelCtrl→en_out_ctrl, 若需要输出则不能为空。 高、宽同 pst_src。	输出
pst_dst_v	由转置后的模板滤波得到的梯度分量图像 V 指针。根据 pstSobelCtrl→en_out_ctrl, 若需要输出则不能为空。 高、宽同 pst_src。	输出
pst_sobel_ctrl	控制信息指针。 不能为空。	输入

参数名称	描述	输入/输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	8x8 ~ 1920x1080
pst_dst_h	S16C1	见【芯片差异】	同 pst_src
pst_dst_v	S16C1	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

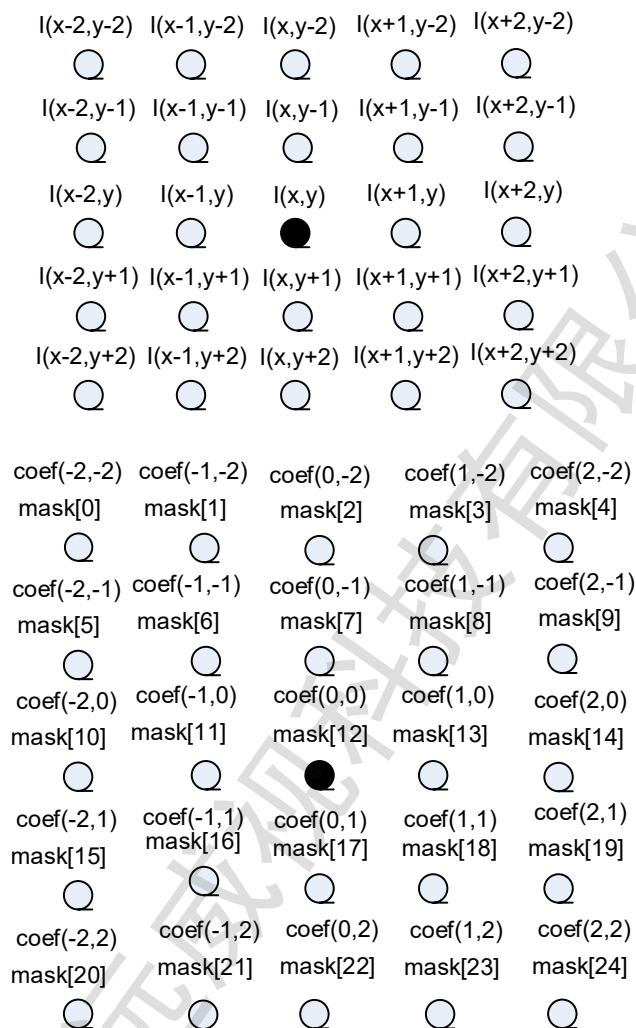
【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【注意】

- 可配置 3 种输出模式，参考 [xmedia_ive_sobel_out_ctrl_e](#)。
- 当输出模式为 XMEDIA_IVE_SOBEL_OUT_CTRL_BOTH 时，要求 pst_dst_h 和 pst_dst_v 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



$$Hout(x, y) = \sum_{-2 < j < 2} \sum_{-2 < i < 2} I(x + i, y + j) \bullet coef(i, j)$$

$$Vout(x, y) = \sum_{-2 < j < 2} \sum_{-2 < i < 2} I(x + i, y + j) \bullet coef(j, i)$$

其中, $I(x, y)$ 对应 pst_src , $Hout(x, y)$ 对应 pst_dst_h , $Vout(x, y)$ 对应 pst_dst_v , $coef(mask)$ 为 pst_sobel_ctrl 中的 $as8_mask[25]$

- sobel 模板

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

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

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

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

- 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

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

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

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

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

【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.a

【举例】

无。

【相关主题】

- [xmedia_ive_magandang](#)
- [xmedia_ive_normgrad](#)

xmedia_ive_magandang

【描述】

创建 5x5 模板梯度幅值与幅角计算任务。

【语法】

```
xmedia_s32 xmedia_ive_magandang(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst_mag,  
xmedia_ive_dst_image_s *pst_dst_ang, xmedia_ive_mag_and_ang_ctrl_s *pst_magandang_ctrl,  
xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst_mag	输出幅值图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_dst_ang	输出幅角图像指针。 根据 pst_mag_and_ang_ctrl→en_out_ctrl, 需要输出则不能为空。 高、宽同 pst_src。	输出
pst_mag_and_ang_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x32 ~ 1920x1080
pst_dst_mag	U16C1	见【芯片差异】	同 pst_src
pst_dst_ang	U8C1	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

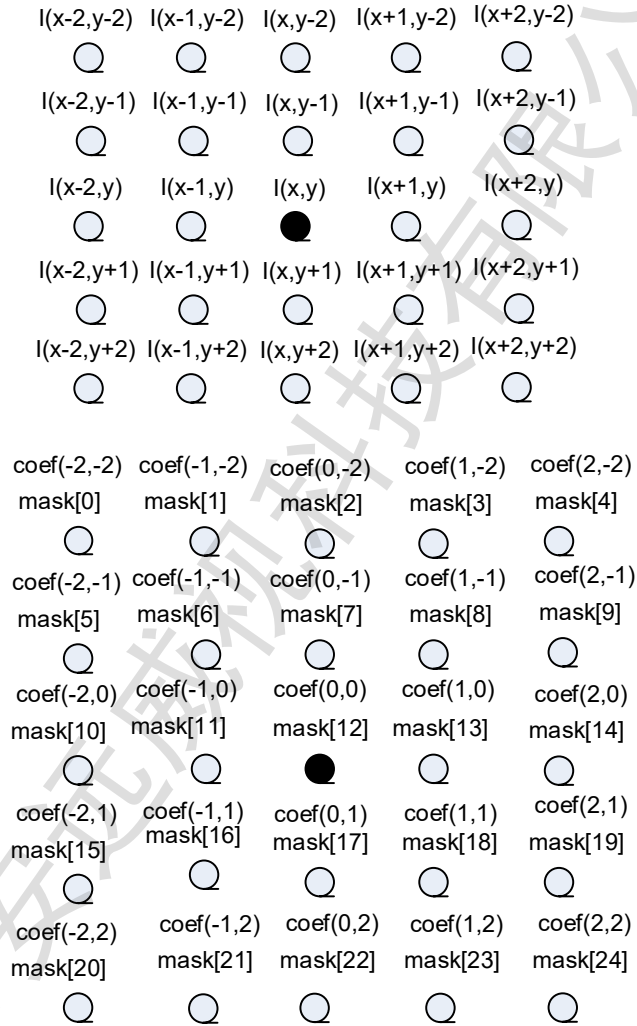
【注意】

- 可配置 2 种输出模式，具体参见 [xmedia_ive_mag_and_ang_out_ctrl_e](#)。
- 当输出模式为 XMEDIA_IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG 时，要求 pst_dst_mag 和 pst_dst_ang 跨度一致。
- 用户可以通过 pst_mag_and_ang_ctrl→u16_thr 对幅值图进行 thresh 操作(可以用来实现 EOH)，计算公式如下：

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

其中, $Mag(x, y)$ 对应 pst_dst_mag 。

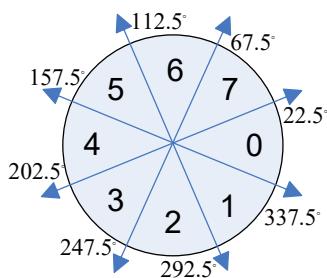
图2-5 MagAndAng 计算示意图



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

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

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



$\theta(x, y)$ 根据 $H_{out}(x, y)$ 、 $V_{out}(x, y)$ 以及 $\arctan(\frac{V_{out}}{H_{out}})$ 取对应上图中 0~7 的方向值。

其中, $I(x, y)$ 对应 `pst_src`, $Mag(x, y)$ 对应 `pst_dst_mag`, $\theta(x, y)$ 对应 `pst_dst_ang`, `coef(mask)` 为 `pst_mag_and_ang_ctrl` 中的 `as8_mask` [25]。

【举例】

无。

【相关主题】

[xmedia_ive_sobel](#)

xmedia_ive_dilate

【描述】

创建二值图像 5x5 模板膨胀任务。

【语法】

```
xmedia_s32 xmedia_ive_dilate(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_dilate_ctrl_s
*pst_dilate_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
<code>p_xmedia_ive_handle</code>	handle 指针。 不能为空。	输出
<code>pst_src</code>	源图像指针。 不能为空。	输入

参数名称	描述	输入/输出
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_dilate_ctrl	控制信息指针。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1 的二值图	见【芯片差异】	32x32 ~ 1920x1080
pst_dst	U8C1 的二值图	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 模板系数只能为 0 或 255。

- 模板样例

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

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

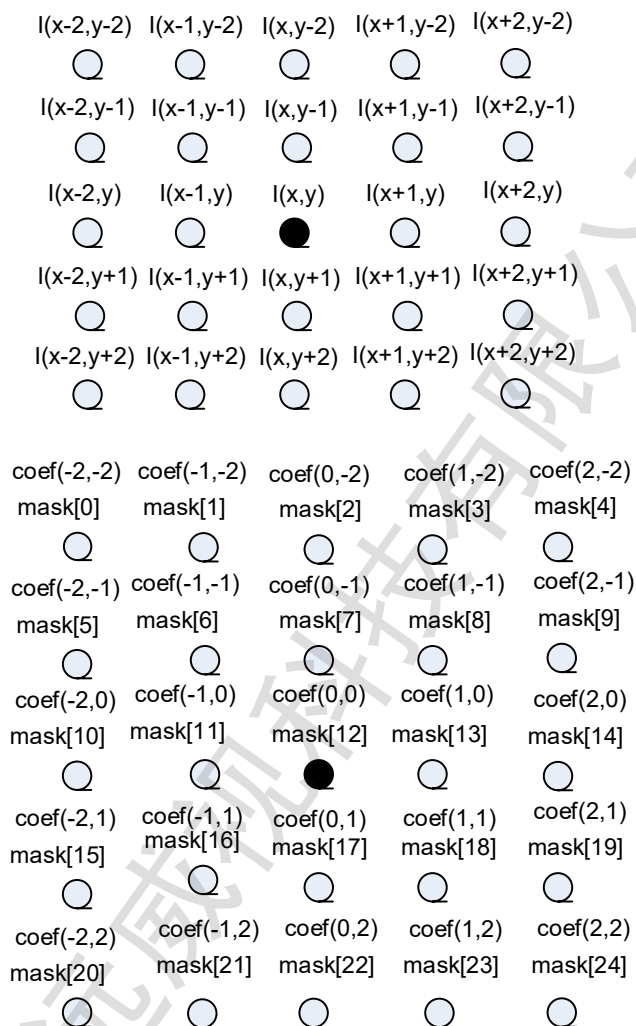
$$\begin{bmatrix} 0 & 255 & 255 & 255 & 0 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 255 & 255 & 255 & 255 & 255 \\ 0 & 255 & 255 & 255 & 0 \end{bmatrix}$$

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

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

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

图2-6 dilate 计算公式示意图



$$I_{out}(x,y) = f(I(x+(k \& 5)-1, y+(k \% 5)-1) \& coef((k \& 5)-1, (k \% 5)-1), |, 0, 24)$$

$$f(A_k, \Theta, c_{min}, c_{max}) = A_{c_{min}} \ominus A_{c_{min}+1} \cdots \ominus A_{c_{max}}$$

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 pst_src, $I_{out}(x,y)$ 对应 pst_dst, $coef(mask)$ 对应 pst_dilate_ctrl 中的 au8_mask[25]。

【举例】

无。

【相关主题】

- [xmedia_ive_erode](#)
- [xmedia_ive_ord_stat_filter](#)

xmedia_ive_erode

【描述】

创建二值图像 5x5 模板腐蚀任务。

【语法】

```
xmedia_s32 xmedia_ive_erode(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_erode_ctrl_s
*pst_erode_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_erode_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1 的二值图	见【芯片差异】	32x32 ~ 1920x1080
pst_dst	U8C1 的二值图	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

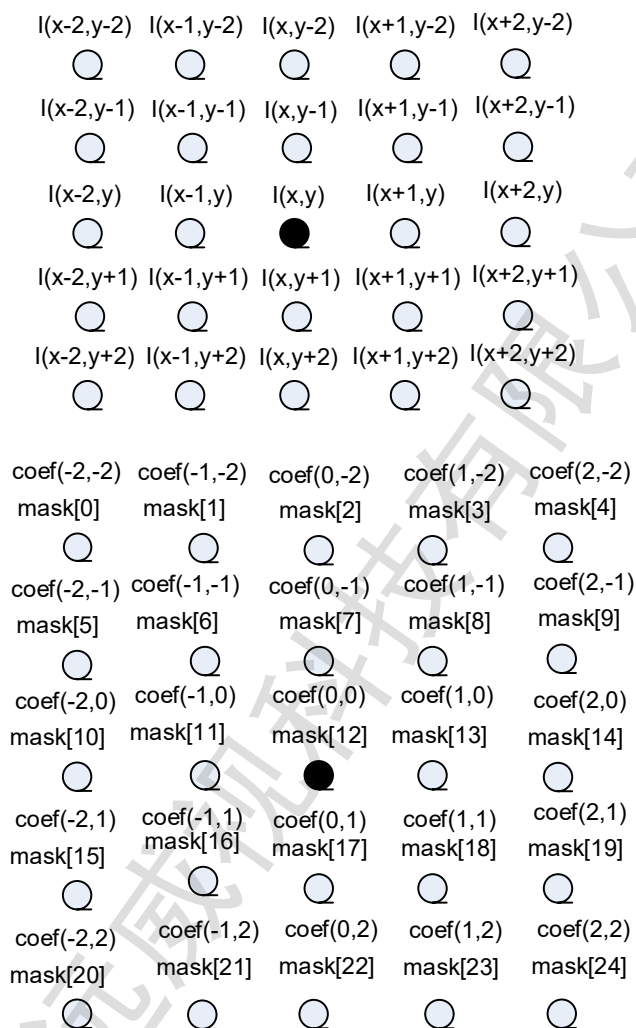
- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 模板系数只能为 0 或 255。
- 模板样例

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

图2-7 erode 计算公式示意图



$$I_{out}(x,y) = f(I(x + (k \& 5) - 1, y + (k \% 5) - 1) | coef((k \& 5) - 1, (k \% 5) - 1), \& 0, 24)$$

$$f(A_k, \Theta, c_{min}, c_{max}) = A_{c_{min}} \Theta A_{c_{min+1}} \cdots \Theta A_{c_{max}}$$

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 `pst_src`， $I_{out}(x,y)$ 对应 `pst_dst`， $coef(mask)$ 对应 `pst_erode_ctrl` 中的 `au8_mask[25]`。

【举例】

无。

【相关主题】

- [xmedia_ive_dilate](#)
- [xmedia_ive_ord_stat_filter](#)

xmedia_ive_thresh

【描述】

创建灰度图像阈值化任务。

【语法】

```
xmedia_s32 xmedia_ive_thresh(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_thresh_ctrl_s
*pst_thr_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_thr_ctrl	控制信息指针。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	1 byte	32x32 ~ 1920x1080
pst_dst	U8C1	1 byte	同 pst_src

【返回值】

返回值	描述
0	成功。

返回值	描述
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可以配置 8 种运算模式，具体参见 [xmedia_ive_thresh_mode_e](#)。
- 计算公式

- XMEDIA_IVE_THRESH_MODE_BINARY:

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

midVal、*highThr* 无需赋值。

- XMEDIA_IVE_THRESH_MODE_TRUNC:

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

minVal、*midVal*、*highThr* 无需赋值。

- XMEDIA_IVE_THRESH_MODE_TO_MINVAL:

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

midVal、*maxVal*、*highThr* 无需赋值。

- XMEDIA_IVE_THRESH_MODE_MIN_MID_MAX:

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

- XMEDIA_IVE_THRESH_MODE_ORI_MID_MAX:

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

minVal 无需赋值。

- XMEDIA_IVE_THRESH_MODE_MIN_MID_ORI:

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

maxVal 无需赋值。

- XMEDIA_IVE_THRESH_MODE_MIN_ORI_MAX:

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

midVal 无需赋值。

- XMEDIA_IVE_THRESH_MODE_ORI_MID_ORI:

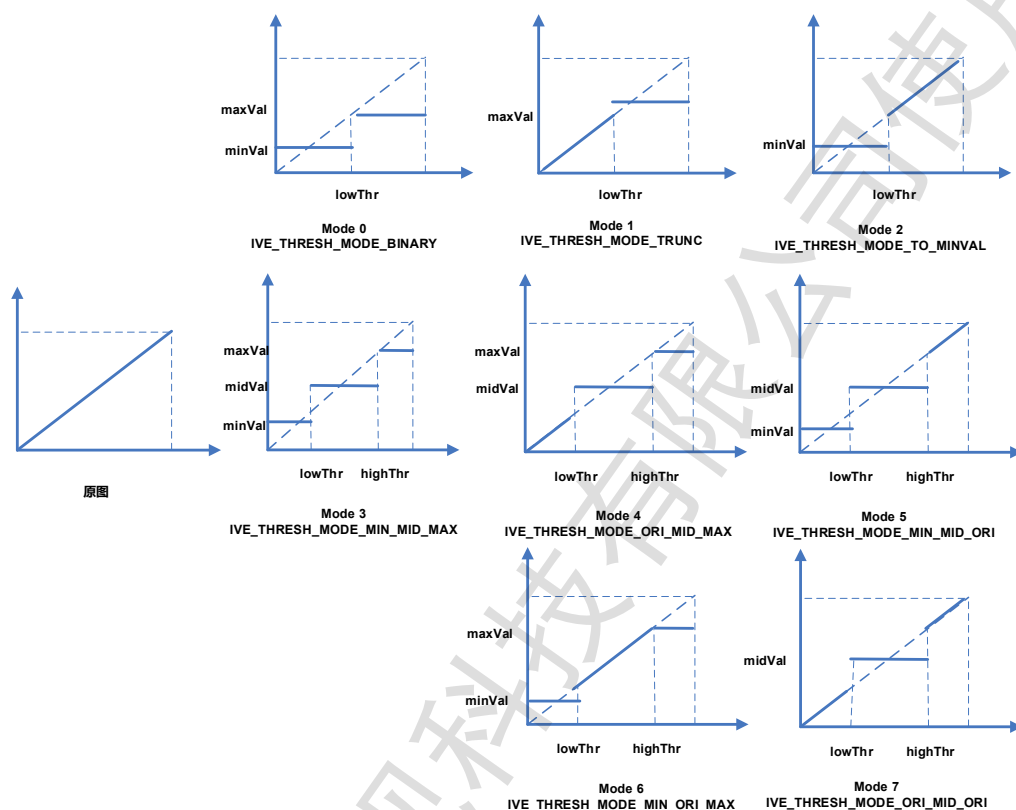
$$I_{out}(x, y) = \begin{cases} I(x, y) & (I(x, y) \leq lowThr) \\ midVal & (lowThr < I(x, y) \leq highThr) \\ I(x, y) & (I(x, y) > highThr) \end{cases}$$

minVal、*maxVal* 无需赋值。

其中, $I(x, y)$ 对应 *pst_src*, $I_{out}(x, y)$ 对应 *pst_dst*, *mode*、*low_thr*、*high_thr*、*min_val*、*mid_val* 和 *max_val* 分别对应 *pst_thr_ctrl* 的 *en_mode*、*u8_low_thr*、*u8_high_thr*、*u8_min_val*、*u8_mid_val* 和 *u8_max_val*。具体示意图如图 2-8 所示。

- *pst_thr_ctrl* 中的 *u8_min_val*、*u8_mid_val* 和 *u8_max_val* 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】

- [xmedia_ive_thresh_s16](#)
- [xmedia_ive_thresh_u16](#)

xmedia_ive_and

【描述】

创建两二值图像相与任务。

【语法】

```
xmedia_s32 xmedia_ive_and(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src1	源图像 1 指针。 不能为空。	输入
pst_src2	源图像 2 指针。 不能为空。 高、宽同 pst_src1。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src1。	输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1 的二值图	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1 的二值图	1 byte	同 pst_src1
pst_dst	U8C1 的二值图	1 byte	同 pst_src1

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

- **【注意】**
- 计算公式如下：
- $I_{out}(x,y) = I_{src1}(x,y) \& I_{src2}(x,y)$
- 其中, $I_{src1}(x,y)$ 对应 pst_src1, $I_{src2}(x,y)$ 对应 pst_src2, $I_{out}(x,y)$ 对应 pst_dst

【举例】

无。

【相关主题】

- [xmedia_ive_or](#)
- [xmedia_ive_xor](#)

xmedia_ive_sub

【描述】

创建两灰度图像相减任务。

【语法】

```
xmedia_s32 xmedia_ive_sub(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_sub_ctrl_s *pst_sub_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。	输出
pst_src1	源图像 1 指针。 不能为空。	输入
pst_src2	源图像 2 指针。 不能为空。 高、宽同 pst_src1。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src1。	输出

参数名称	描述	输入/输出
pst_sub_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1	1 byte	同 pst_src1
pst_dst	U8C1、S8C1	1 byte	同 pst_src1

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可以配置 2 种输出格式，具体参见 [xmedia_ive_sub_mode_e](#)。
- XMEDIA_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
- XMEDIA_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 pst_src1, $I_{src2}(x,y)$ 对应 pst_src2, $I_{out}(x,y)$ 对应 pst_dst。

【举例】

无。

【相关主题】

[xmedia_ive_add](#)

xmedia_ive_or

【描述】

创建两二值图像相或任务。

【语法】

```
xmedia_s32 xmedia_ive_or(xmedia_ive_handle *p_xmedia_ive_handle, xmedia\_ive\_src\_image\_s
*pst_src1, xmedia\_ive\_src\_image\_s *pst_src2, xmedia\_ive\_dst\_image\_s *pst_dst, xmedia_bool
b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src1	源图像 1 指针。 不能为空。	输入
pst_src2	源图像 2 指针。 不能为空。 高、宽同 pst_src1。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src1。	输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1	1 byte	同 pst_src1
pst_dst	U8C1	1 byte	同 pst_src1

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.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 pst_src1， $I_{src2}(x, y)$ 对应 pst_src2， $I_{out}(x, y)$ 对应 pst_dst。

【举例】

无。

【相关主题】

- [xmedia_ive_and](#)
- [xmedia_ive_xor](#)

xmedia_ive_integ

【描述】

创建灰度图像的积分图计算任务。

【语法】

```
xmedia_s32 xmedia_ive_integ(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_integ_ctrl_s  
*pst_integ_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_integ_ctrl	控制信息指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x16 ~ 1920x1080
pst_dst	U32C1、U64C1	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- XMEDIA_IVE_INTEG_OUT_CTRL_COMBINE，组合输出模式，输出图像类型必须为 XMEDIA_IVE_IMAGE_TYPE_U64C1，计算公式如下：

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

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

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

- XMEDIA_IVE_INTEG_OUT_CTRL_SUM，仅和积分图输出模式，输出图像类型必须为 XMEDIA_IVE_IMAGE_TYPE_U32C1，计算公式如下：

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

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

- XMEDIA_IVE_INTEG_OUT_CTRL_SQSUM，仅平方和积分图输出，输出图像类型必须为 XMEDIA_IVE_IMAGE_TYPE_U64C1，计算公式如下：

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

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

其中， $I(x, y)$ 对应 pst_src， $I_{out}(x, y)$ 对应 pst_dst。

【举例】

无。

【相关主题】

无。

xmedia_ive_hist

【描述】

创建灰度图像的直方图统计任务。

【语法】

```
xmedia_s32 xmedia_ive_hist(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s
*pst_src, xmedia_ive_dst_mem_info_s *pst_dst, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出数据指针。 不能为空。 内存至少配置 1024 字节。	输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x32 ~ 1920x1080
pst_dst	-	见【芯片差异】	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.a

【注意】

计算公式如下:

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

其中, $I(i, j)$ 对应 pst_src, $I_{out}(x)$ 对应 pst_dst。

【举例】

无。

【相关主题】

无。

xmedia_ive_thresh_s16

【描述】

创建 S16 数据到 8bit 数据的阈值化任务。

【语法】

```
xmedia_s32 xmedia_ive_thresh_s16(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,  
xmedia_ive_thresh_s16_ctrl_s *pst_thr_s16_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_thr_s16_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	S16C1	2 byte	32x32 ~ 1920x1080
pst_dst	U8C1、S8C1	1 byte	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可配置 4 种运算模式，参考 [xmedia_ive_thresh_s16_mode_e](#)。
- 计算公式

- XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX:

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

要求: $-32768 \leq lowThr \leq highThr \leq 32767$;

$-128 \leq minVal$ 、 $midVal$ 、 $maxVal \leq 127$ 。

- XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX:

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

要求: $-129 \leq lowThr \leq highThr \leq 127$;

$-128 \leq minVal$ 、 $maxVal \leq 127$;

- XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX:

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

要求: $-32768 \leq lowThr \leq highThr \leq 32767$;

$0 \leq minVal$ 、 $midVal$ 、 $maxVal \leq 255$ 。

- XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX:

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

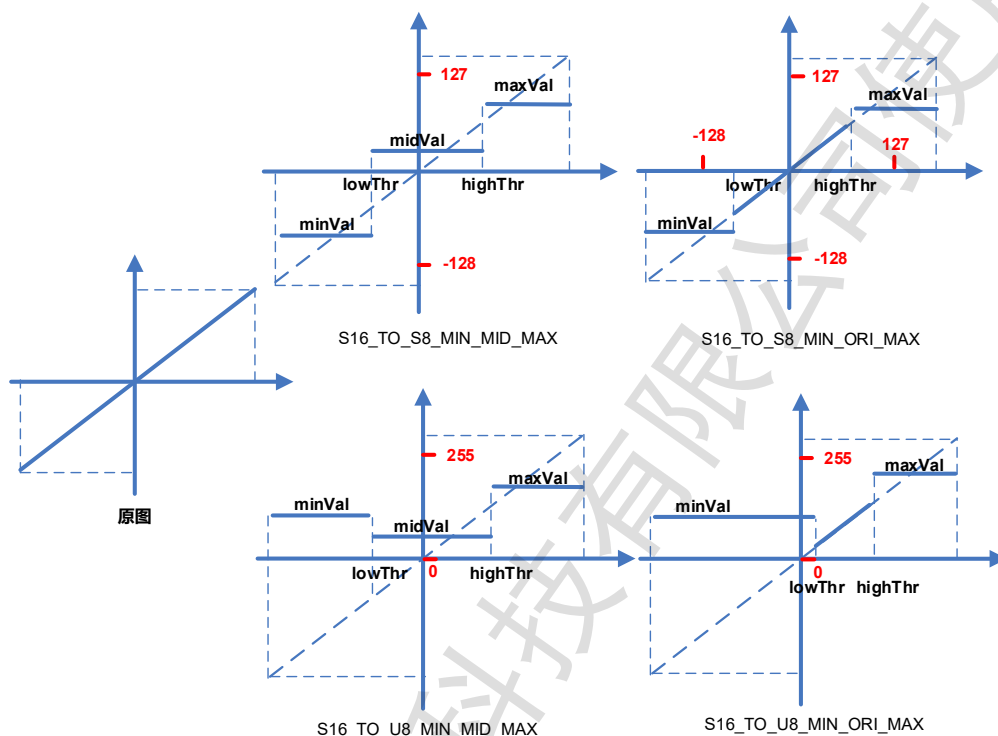
要求: $-1 \leq lowThr \leq highThr \leq 255$;

$0 \leq minVal$ 、 $maxVal \leq 255$ 。

其中, $I(x, y)$ 对应 pst_src , $I_{out}(x, y)$ 对应 pst_dst , $mode$ 、 low_thr 、 $high_thr$ 、 min_val 、 mid_val 和 $maxVal$ 分别对应 $pst_thr_s16_ctrl$ 的 en_mode 、 $s16_low_thr$ 、 $s16_high_thr$ 、 $un8_min_vVal$ 、 $un8_mid_val$ 和 $un8_max_val$ 。具体示意图如图 2-9 所示。

- $pst_thr_s16_ctrl$ 中的 $un8_min_val$ 、 $un8_mid_val$ 和 $un8_max_val$ 并不需要满足变量命名含义中的大小关系。

图2-9 thresh_S16 4 种阈值化模式示意图



【举例】

无。

【相关主题】

- [xmedia_ive_thresh_u16](#)
- [xmedia_ive_16bit_to_8bit](#)

xmedia_ive_thresh_u16

【描述】

创建 U16 数据到 U8 数据的阈值化任务。

【语法】

```
xmedia_s32 xmedia_ive_thresh_u16(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_thresh_u16_ctrl_s *pst_thr_u16_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_thr_u16_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U16C1	2 byte	16x16 ~ 1920x1080
pst_dst	U8C1	1 byte	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可配置 2 种运算模式，参考 [xmedia_ive_thresh_u16_mode_e](#)。

- 计算公式

- XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX:

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

要求: $0 \leq lowThr \leq highThr \leq 65535$;

- XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX:

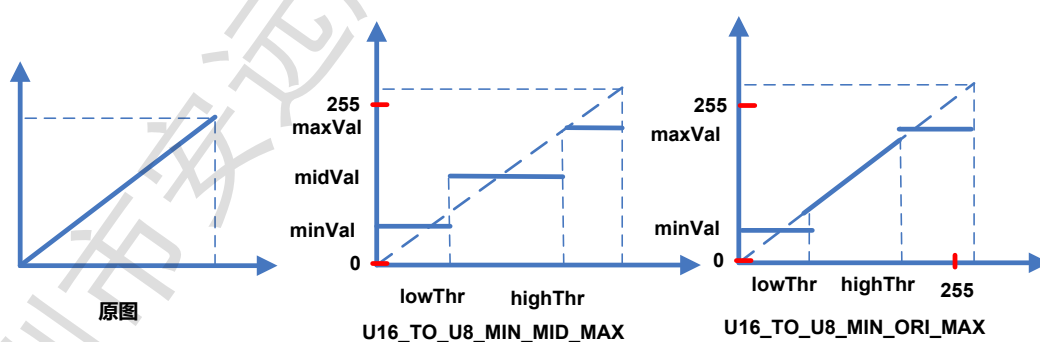
$$I_{out}(x,y) = \begin{cases} minVal & (I(x,y) \leq lowThr) \\ I(x,y) & (lowThr < I(x,y) \leq highThr) \\ maxVal & (I(x,y) > highThr) \end{cases}$$

要求: $0 \leq lowThr \leq highThr \leq 255$;

其中, $I(x,y)$ 对应 `pst_src`, $I_{out}(x,y)$ 对应 `pst_dst`, `mode`、`low_Thr`、`high_Thr`、`min_Val`、`mid_val` 和 `max_val` 分别对应 `pst_thr_u16_ctrl` 的 `en_mode`、`u16_low_thr`、`u16_high_thr`、`u8_min_val`、`u8_mid_val` 和 `u8_max_val`。具体示意图如图 2-10 所示。

- `pst_thr_u16_ctrl` 中的 `u8_min_val`、`u8_mid_val` 和 `u8_max_val` 并不需要满足变量命名含义中的大小关系。

图2-10 thresh_U16 2 种阈值化模式示意图



【举例】

无。

【相关主题】

- [xmedia_ive_thresh_s16](#)

- [xmedia_ive_16bit_to_8bit](#)

xmedia_ive_16bit_to_8bit

【描述】

创建 16bit 图像数据到 8bit 图像数据的线性转化任务。

【语法】

```
xmedia_s32 xmedia_ive_16bit_to_8bit(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_16bit_to_8bit_ctrl_s *pst_16bit_to_8bit_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_16bit_to_8bit_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

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

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可配置 4 种模式，具体参考 [xmedia_ive_16bit_to_8bit_mode_e](#)。
- 计算公式

- XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8:

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

- XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS:

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

- XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS:

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

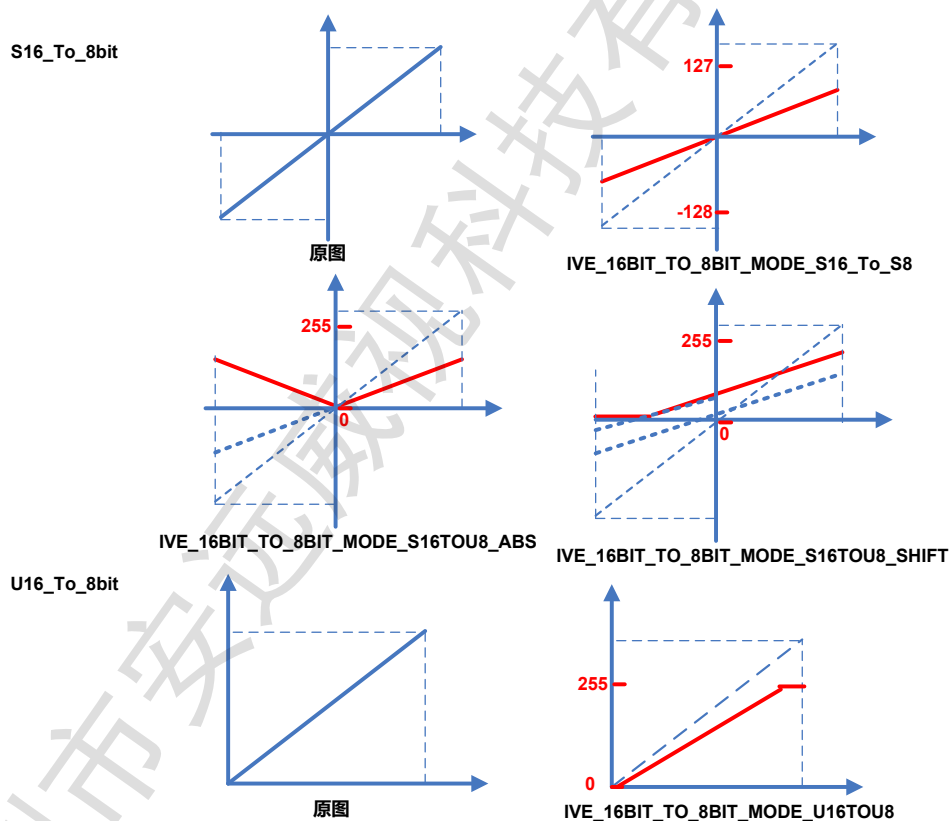
- XMEDIA_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8:

$$I_{out}(x,y)=\begin{cases} 0 & (\frac{a}{b}I(x,y)<0) \\ \frac{a}{b}I(x,y) & (0\leq\frac{a}{b}I(x,y)\leq255) \\ 255 & (\frac{a}{b}I(x,y)>255) \end{cases}$$

其中， $I(x,y)$ 对应 pst_src， $I_{out}(x,y)$ 对应 pst_dst，mode、 a 、 b 和 $bias$ 分别对应 pst_16bit_to_8bit_ctrl 的 en_mode、u8_numerator、u16_denominator、s8_bias。具体示意图如图 2-11 所示。

要求：u8_numerator ≤ u16_denominator，且 u16_denominator ≠ 0。

图2-11 16BitTo8Bit 4 种转换模式示意图



【举例】

无。

【相关主题】

- [xmedia_ive_thresh_s16](#)
- [xmedia_ive_thresh_u16](#)

xmedia_ive_ord_stat_filter

【描述】

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

【语法】

```
xmedia_s32 xmedia_ive_ord_stat_filter(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,  
xmedia_ive_ord_stat_filter_ctrl_s *pst_ordstatflt_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_ord_statflt_ctrl	控制参数指针 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x32 ~ 1920x1080
pst_dst	U8C1	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 可配置 3 种滤波模式，参考 [xmedia_ive_ord_stat_filter_mode_e](#)。
- 计算公式

- XMEDIA_IVE_ORD_STAT_FILTER_MODE_MEDIAN:

$$I_{out}(x, y) = \underset{\substack{-1 \leq i \leq 1 \\ -1 \leq j \leq 1}}{\text{median}} \{I(x+i, y+j)\}$$

- XMEDIA_IVE_ORD_STAT_FILTER_MODE_MAX:

$$I_{out}(x, y) = \underset{\substack{-1 \leq i \leq 1 \\ -1 \leq j \leq 1}}{\text{max}} \{I(x+i, y+j)\}$$

- XMEDIA_IVE_ORD_STAT_FILTER_MODE_MIN:

$$I_{out}(x, y) = \underset{\substack{-1 \leq i \leq 1 \\ -1 \leq j \leq 1}}{\text{min}} \{I(x+i, y+j)\}$$

其中， $I(x, y)$ 对应 pst_src， $I_{out}(x, y)$ 对应 pst_dst。

【举例】

无。

【相关主题】

- [xmedia_ive_filter](#)
- [xmedia_ive_dilate](#)

- [xmedia_ive_erode](#)

xmedia_ive_map

【描述】

创建 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、U8C1→U16C1、U8C1→S16C1 3 种模式的映射。

【语法】

```
xmedia_s32 xmedia_ive_map(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia\_ive\_src\_image\_s *pst_src, xmedia\_ive\_src\_mem\_info\_s *pst_map,
xmedia\_ive\_dst\_image\_s *pst_dst, xmedia\_ive\_map\_ctrl\_s *pst_map_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_map	映射表信息指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_map_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x32 ~ 1920x1080

参数名称	支持图像类型	地址对齐	分辨率
pst_map	-	见【芯片差异】	-
pst_dst	U8C1、U16C1、S16C1	见【芯片差异】	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libive.a (PC 上模拟用 ive_clib2.1.lib)

【注意】

- 计算公式如下：

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

其中， $I(x, y)$ 对应 pst_src， $I_{out}(x, y)$ 对应 pst_dst，map 对应 pst_map。

- pstMap 的内存配置根据 pst_map_ctrl→en_mode 配置不同：
 - XMEDIA_IVE_MAP_MODE_U8，配置 sizeof(xmedia_ive_map_u8bit_lut_mem_s);
 - XMEDIA_IVE_MAP_MODE_U16，配置 sizeof(xmedia_ive_map_u16bit_lut_mem_s);

- XMEDIA_IVE_MAP_MODE_S16, 配置 sizeof(xmedia_ive_map_s16bit_lut_mem_s).

【举例】

无。

【相关主题】

无。

xmedia_ive_equalize_hist

【描述】

创建灰度图像的直方图均衡化计算任务。

【语法】

```
xmedia_s32 xmedia_ive_equalizehist(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_equalize_hist_ctrl_s *pst_equalize_hist_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_equalize_hist_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1	见【芯片差异】	32x32 ~ 1920x1080
pst_dst	U8C1	见【芯片差异】	同 pst_src
pst_equalize_hist_ctrl →st_mem	-	见【芯片差异】	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- pst_equalize_hist_ctrl 中的 st_mem，至少需开辟 sizeof(xmedia_ive_equalize_hist_ctrl_mem_s) 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。

xmedia_ive_add

【描述】

创建两灰度图像的加权加计算任务。

【语法】

```
xmedia_s32 xmedia_ive_add(xmedia_ive_handle *p_xmedia_ive_handle,  
xmedia_ive_src_image_s *pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s  
*pst_dst, xmedia_ive_add_ctrl_s *pst_add_ctrl, xmedia_bool b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src1	源图像 1 指针。 不能为空。	输入
pst_src2	源图像 2 指针。 不能为空。 高、宽同 pst_src1。	输入
pst_dst	输出图像指针。 高、宽同 pst_src1; 不能为空。	输出
pst_add_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1	1 byte	同 pst_src
pst_dst	U8C1	1 byte	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

计算公式如下：

$I_{out}(i, j) = x * I_1(i, j) + y * I_2(i, j)$ 其中， $I_1(i, j)$ 对应 pst_src1， $I_2(i, j)$ 对应 pst_src2， $I_{out}(i, j)$ 对应 pst_dst； x ， y 为 pst_add_ctrl 中的 u0q16X，u0q16Y；若定点化前， x 和 y 满足 $x + y > 1$ ，则当计算结果超过 8bit 取低 8bit 作为最终结果。

【举例】

无。

【相关主题】

[xmedia_ive_sub](#)

xmedia_ive_xor

【描述】

创建两二值图的异或计算任务。

【语法】

```
xmedia_s32 xmedia_ive_xor(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst, xmedia_bool
b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src1	源图像 1 指针。 不能为空。	输入
pst_src2	源图像 1 指针。 不能为空。 高、宽同 pst_src1。	输入
pst_dst	输出图像指针。 不能为空。 高、宽同 pst_src1。	输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1	1 byte	同 pst_src
pst_dst	U8C1	1 byte	同 pst_src

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.a

【注意】

计算公式如下：

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

其中, $I_{src1}(x, y)$ 对应 pst_src1, $I_{src2}(x, y)$ 对应 pst_src2, $I_{dst}(x, y)$ 对应 pst_dst。

【举例】

无。

【相关主题】

- [xmedia_ive_and](#)
- [xmedia_ive_or](#)

xmedia_ive_ncc

【描述】

创建两相同分辨率灰度图像的归一化互相关系数计算任务。

【语法】

```
xmedia_s32 xmedia_ive_ncc(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s  
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_mem_info_s *pst_dst, xmedia_bool  
b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src1	源 1 图像指针。 不能为空。	输入
pst_src2	源 2 图像指针。 不能为空。 高、宽同 pst_src1。	输入

参数名称	描述	输入/输出
pst_dst	输出数据指针。 不能为空。 内存至少需配置：sizeof (xmedia_ive_ncc_dst_mem_s)。	输出
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src1	U8C1	1 byte	32x32 ~ 1920x1080
pst_src2	U8C1	1 byte	同 pst_src
pst_dst	-	16 byte	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	支持
XM7206V11A	不支持

【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- 计算公式如下

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

- 仅输出上面公式的分子、开方之前的两个分母项，即 pst_dst→u64_numerator、pst_dst→u64_quad_sum1、pst_dst→u64_quad_sum2 分别对应上面公式的

$$\sum_{i=1}^w \sum_{j=1}^h (I_{src1}(i, j) * I_{src2}(i, j))、\sum_{i=1}^w \sum_{j=1}^h (I_{src1}^2(i, j))、\sum_{i=1}^w \sum_{j=1}^h (I_{src2}^2(i, j))。$$

【举例】

无。

【相关主题】

无。

xmedia_ive_ccl

【描述】

创建二值图像的连通区域标记任务。

【语法】

```
xmedia_s32 xmedia_ive_ccl(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_image_s
*pst_src_dst, xmedia_ive_dst_mem_info_s *pst_blob, xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl,
xmedia_bool b_instant);
```

【参数】

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

参数名称	描述	输入/输出
pst_blob	连通区域信息指针。 不能为空。 内存至少需配置为 sizeof (xmedia_ive_ccblob_s)大小，最多输出 254 个有效的连通区域。	输出
pst_ccl_ctrl	控制参数指针 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src_dst	U8C1	见【芯片差异】	16x16 ~ 1280x720
pst_blob	-	见【芯片差异】	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

【需求】

- 头文件: xmedia_ive_common.h、xmedia_ive.h
- 库文件: libxmedia_ive.a

【注意】

- 连通区域的信息保存在 pst_blob→ast_region 中。
- pst_blob→u8_region_num 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 pst_blob→u16_cur_area_thr, 标记号为其所在 pst_blob→ast_region 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 pst_blob→s8_label_status 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用一个标记)。对于后者, 若用户需要正确的标记号, 还需要再次根据 pst_blob 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 pst_ccl_ctrl→u16_init_area_thr 进行筛选, 面积小于等于 pst_ccl_ctrl→u16_init_area_thr 均会被置为 0。
- 当连通区域数目大于 254, 会用 pst_ccl_ctrl→u16_init_area_thr 删除面积小的连通区域; 若 pst_ccl_ctrl→u16_init_area_thr 不满足删除条件, 会以 pst_ccl_ctrl→u16_step 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 pst_blob→u16_init_area_thr 中。

【举例】

无。

【相关主题】

无。

xmedia_ive_gmm

【描述】

创建 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】

```
xmedia_s32 xmedia_ive_gmm(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_fg, xmedia_ive_dst_image_s
*pst_bg, xmedia_ive_mem_info_s *pst_model, xmedia_ive_gmm_ctrl_s *pst_gmm_ctrl,
```

xmedia_bool b_instant);

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_fg	前景图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_bg	背景图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_model	GMM 模型参数指针。 不能为空。	输入、输出
pst_gmm_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1、U8C3_PACKAGE	16 byte	32x32 ~ 1920x1080
pst_fg	U8C1 的二值图	同 pst_src	同 pst_src
pst_bg	同 pst_src	同 pst_src	同 pst_src
pst_model	-	同 pst_src	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	支持
XM7206V11A	不支持

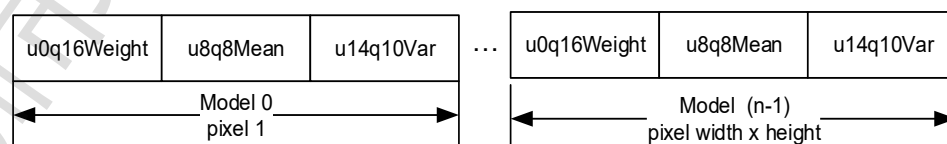
【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.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，pst_model 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

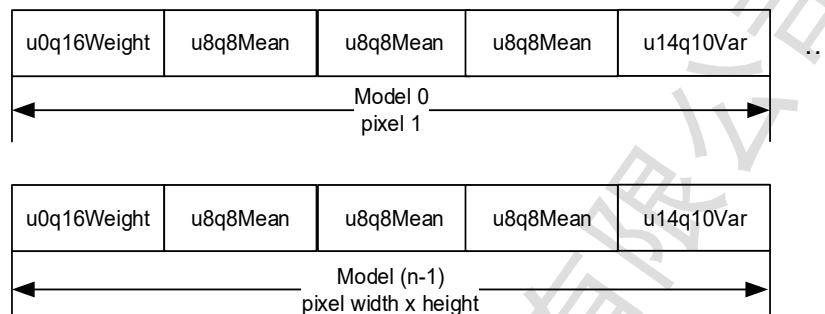


一个像素的单个高斯模型参数 weight 用 2 字节、mean 用 2 字节、var 用 3 字节；因此 pst_model 需要分配的内存大小：

```
pst_model->u32_size = 7 * pst_src->u16_width * pst_src->u16_height *
pst_gmm_ctrl->u8_mode_num
```

- RGB 图像 GMM 采用 n 个(n=3 或 5)高斯模型，pst_model 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 2 字节、mean[3]用 2*3 字节、var 用 3 字节；因此 pst_model 需要分配的内存大小：

```
pst_model→u32Size = 11 * pst_src→u16_width * pst_src→u16_height *
pst_gmm_ctrl→u8_mode_num
```

【举例】

无。

【相关主题】

- [xmedia_ive_match_bg_model](#)
- [xmedia_ive_update_bg_model](#)
- [xmedia_ive_gmm2](#)

xmedia_ive_gmm2

【描述】

创建 GMM 背景建模任务，支持 1-5 个高斯模型，支持灰度图和 RGB_PACKAGE 图输入，支持全局及像素级别的灵敏度系数以及前景模型时长更新系数。

【语法】

```
xmedia_s32 xmedia_ive_gmm2(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_src_image_s *pst_factor, xmedia_ive_dst_image_s
*pst_fg, xmedia_ive_dst_image_s *pst_bg, xmedia_ive_dst_image_s *pst_match_model_info,
xmedia_ive_mem_info_s *pst_model, xmedia_ive_gmm2_ctrl_s *pst_gmm2_ctrl, xmedia_bool
b_instant);
```

【参数】

参数名称	描述	输入/输出
p_xmedia_ive_handle	handle 指针。 不能为空。	输出
pst_src	源图像指针。 不能为空。	输入
pst_factor	模型更新参数指针。 当且仅当 pst_gmm2_ctrl→ en_sns_factor_mode pst_gmm2_ctrl→ en_life_update_factor_mode 均使用全局模 式时可以为空。	输入
pst_fg	前景图像指针。 不能为空。 高、宽同 pst_src。	输出
pstbg	背景图像指针。 不能为空。 高、宽同 pst_src。	输出
pst_match_model_info	模型匹配系数指针。 不能为空。	输出
pst_model	GMM 模型参数指针。 不能为空。	输入、输出
pst_gmm2_ctrl	控制参数指针。 不能为空。	输入
b_instant	及时返回结果标志。	输入

参数名称	支持图像类型	地址对齐	分辨率
pst_src	U8C1、 U8C3_PACKAGE	见【芯片差异】	32x32 ~ 1920x1080
pst_factor	U16C1	见【芯片差异】	同 pst_src
pst_fg	U8C1 的二值图	见【芯片差异】	同 pst_src
pst_bg	同 pst_src	见【芯片差异】	同 pst_src
pst_match_model_info	U8C1	见【芯片差异】	同 pst_src
pst_model	-	见【芯片差异】	-

【返回值】

返回值	描述
0	成功。
非 0	失败，参见错误码。

【芯片差异】

芯片类型	差异
XM7606V13	地址为 16 字节对齐
XM7206V11A	地址为 8 字节对齐

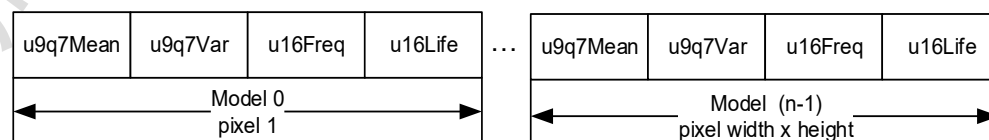
【需求】

- 头文件：xmedia_ive_common.h、xmedia_ive.h
- 库文件：libxmedia_ive.a

【注意】

- gmm2 在参考了 OPENCV 的 MOG 和 MOG2 的基础上，增加了像素级别的参数控制。
- 源图像 pst_src 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。
- 模型更新参数 pst_factor 为 U16C1 图像：每个元素用 16 bit 表示，低 8 bit 为灵敏度系数，用于控制模型匹配时方差倍数；高 8 bit 为前景模型时长更新参数，用于控制背景模型形成时间。
- 模型匹配系数指针 pst_match_model_info 为 U8C1 图像：每个元素用 8bit 表示，低 1 bit 为高斯模型匹配标志，0 表示匹配失败，1 表示匹配成功；高 7 bit 为频率最大模型序号。
- gmm2 的频率参数 (pst_gmm2_ctrl 中的 u16_freq_init_val、u16_freq_redu_factor、u16_freq_add_factor、u16_freq_thr) 用于控制模型排序和模型有效时间。
 - u16_freq_init_val 越大，模型有效时间越大；
 - u16_freq_redu_factor 越大，模型有效时间越长，模型频率通过乘以频率衰减系数 $u16_freq_redu_factor/65536$ ，达到频率衰减的目的；
 - u16_freq_add_factor 越大，模型有效时间越长；
 - u16_freq_thr 越大，模型有效时间越短。
- gmm2 的模型时长参数(pst_gmm2_ctrl 中的 u16_life_thr)用于控制前景模型成为背景的时间。
 - u16_life_thr 越大，前景持续时间越长；
 - 单高斯模型下，模型时长参数不生效。
- 灰度图像 gmm2 采用 n 个 ($1 \leq n \leq 5$) 高斯模型，pst_model 的内存排列方式如图 2-14 所示。

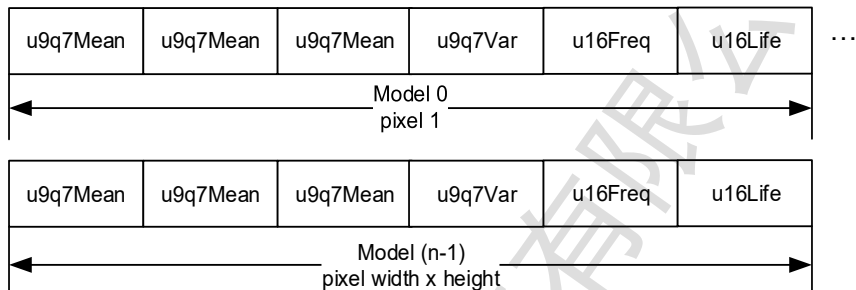
图2-14 灰度图像 gmm2 模型的内存配置示意图



一个像素的单个高斯模型参数 mean 用 2 字节，var 用 2 字节，freq 用 2 字节，life 用 2 字节；因此 pst_model 需要分配的内存大小：

- ```
pst_model->u32_size = 8 * pst_src->u16_width*pst_src->u16_height *
pst_gmm2_ctrl->u8_model_num
```
- RGB 图像 gmm2 采用  $n$  个 ( $1 \leq n \leq 5$ ) 高斯模型，pst\_model 的内存排列方式如图 2-15 所示。

图2-15 RGB 图像 gmm2 模型的内存配置示意图



一个像素的单个高斯模型参数 mean[3]用 6 字节，var 用 2 字节，freq 用 2 字节，life 用 2 字节；因此 pst\_model 需要分配的内存大小：

```
pst_model->u32_size = 12 * pst_src->u16_width*pst_src->u16_height *
pst_gmm2_ctrl->u8_model_num
```

#### 【举例】

无。

#### 【相关主题】

- [xmedia\\_ive\\_match\\_bg\\_model](#)
- [xmedia\\_ive\\_update\\_bg\\_model](#)
- [xmedia\\_ive\\_gmm](#)

## xmedia\_ive\_normgrad

#### 【描述】

创建归一化梯度计算任务，梯度分量均归一化到 S8。

#### 【语法】

```
xmedia_s32 xmedia_ive_normgrad(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src,xmedia_ive_dst_image_s *pst_dst_h, xmedia_ive_dst_image_s
*pst_dst_v, xmedia_ive_dst_image_s *pst_dst_hv, xmedia_ive_norm_grad_ctrl_s
*pst_normgrad_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                                                                                   | 输入/输出 |
|---------------------|------------------------------------------------------------------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                                                                                  | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                                                                                      | 输入    |
| pst_dst_h           | 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。<br>根据 pstNormGradCtrl→en_out_ctrl, 若需要输出则不能为空。                       | 输出    |
| pst_dst_v           | 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。<br>根据 pst_norm_grad_ctrl→en_out_ctrl, 若需要输出则不能为空。                  | 输出    |
| pst_dst_hv          | 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 格式存储的图像指针。<br>根据 pst_norm_grad_ctrl→en_out_ctrl, 若需要输出则不能为空。 | 输出    |
| pst_norm_grad_ctrl  | 控制信息指针。                                                                                              | 输入    |
| b_instant           | 及时返回结果标志。                                                                                            | 输入    |

| 参数名称      | 支持图像类型 | 地址对齐    | 分辨率               |
|-----------|--------|---------|-------------------|
| pst_src   | U8C1   | 见【芯片差异】 | 32x32 ~ 1920x1080 |
| pst_dst_h | S8C1   | 见【芯片差异】 | 同 pst_src         |
| pst_dst_v | S8C1   | 见【芯片差异】 | 同 pst_src         |

| 参数名称       | 支持图像类型       | 地址对齐    | 分辨率       |
|------------|--------------|---------|-----------|
| pst_dst_hv | S8C2_PACKAGE | 见【芯片差异】 | 同 pst_src |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异          |
|------------|-------------|
| XM7606V13  | 地址为 16 字节对齐 |
| XM7206V11A | 地址为 8 字节对齐  |

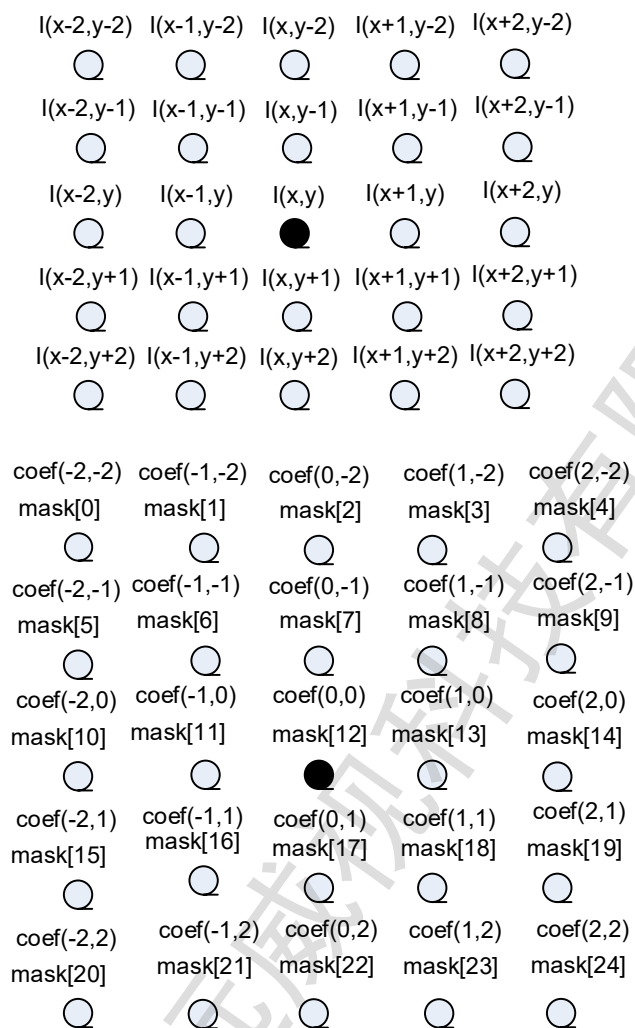
#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

- 控制参数中输出模式如下：
  - XMEDIA\_IVE\_NORM\_GRAD\_OUT\_CTRL\_HOR\_AND\_VER 时，pst\_dst\_h 和 pst\_dst\_v 指针不能为空，且要求跨度一致；
  - XMEDIA\_IVE\_NORM\_GRAD\_OUT\_CTRL\_HOR 时，pst\_dst\_h 不能为空；
  - XMEDIA\_IVE\_NORM\_GRAD\_OUT\_CTRL\_VER 时，pst\_dst\_v 不能为空；
  - XMEDIA\_IVE\_NORM\_GRAD\_OUT\_CTRL\_COMBINE 时，pst\_dst\_hv 不能为空。
- NormGrad 计算公式如图 2-16 所示。

图2-16 NormGrad 计算公式示意图



$$I_{out}(x, y) = \left\{ \sum_{-2 < j < 2} \sum_{-2 < i < 2} I(x+i, y+j) \bullet coef(i, j) \right\} \gg \text{norm}$$

【举例】

无。

【相关主题】

[xmedia\\_ive\\_sobel](#)

[xmedia\\_ive\\_st\\_candi\\_corner](#)

【描述】

灰度图像 Shi-Tomasi-like 角点计算的前半部：计算候选角点。

### 【语法】

```
xmedia_s32 xmedia_ive_st_candi_corner(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_candi_corner,
xmedia_ive_st_candi_corner_ctrl_s *pst_st_candi_corner_ctrl, xmedia_bool b_instant);
```

### 【参数】

| 参数名称                     | 描述                                     | 输入/输出 |
|--------------------------|----------------------------------------|-------|
| p_xmedia_ive_handle      | handle 指针。<br>不能为空。                    | 输出    |
| pst_src                  | 源图像指针。<br>不能为空。                        | 输入    |
| pst_candi_corner         | 候选角点响应值图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_st_candi_corner_ctrl | 控制参数指针。<br>不能为空。                       | 输入    |
| b_instant                | 及时返回结果标志。                              | 输入    |

| 参数名称                                | 支持图像类型 | 地址对齐    | 分辨率            |
|-------------------------------------|--------|---------|----------------|
| pst_src                             | U8C1   | 见【芯片差异】 | 16x16~1280x720 |
| pst_candi_corner                    | U8C1   | 见【芯片差异】 | 同 pst_src      |
| pst_st_candi_corner_ctrl<br>→st_mem | -      | 见【芯片差异】 | -              |

### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异          |
|------------|-------------|
| XM7606V13  | 地址为 16 字节对齐 |
| XM7206V11A | 地址为 8 字节对齐  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

- 与 OpenCV 中 ShiTomas 角点计算原理类似。
- pst\_st\_candi\_corner\_ctrl→st\_mem 至少需开辟的内存大小：  

$$\begin{aligned} & \text{pst\_st\_candi\_corner\_ctrl} \rightarrow \text{st\_mem.u32\_size} \\ &= 4 * \text{IveGetStride}(\text{pst\_src} \rightarrow \text{u16\_width}, \text{XMEDIA\_IVE\_STRIDE\_ALIGN}) * \\ & \text{pst\_src} \rightarrow \text{u16\_height} \\ &+ \text{sizeof}(\text{xmedia\_ive\_st\_max\_eig\_s})。 \end{aligned}$$
- 该任务完成后，必须要使用 [xmedia\\_ive\\_st\\_corner](#) 函数才能得到真正的角点。

#### 【举例】

无。

#### 【相关主题】

[xmedia\\_ive\\_st\\_corner](#)

xmedia\_ive\_st\_corner

#### 【描述】

灰度图像 Shi-Tomasi-like 角点计算的后半部：按规则挑选角点。

#### 【语法】

```
xmedia_s32 xmedia_ive_st_corner(xmedia_ive_src_image_s *pst_candi_corner,
xmedia_ive_dst_mem_info_s *pst_corner, xmedia_ive_st_corner_ctrl_s *pst_st_corner_ctrl);
```

#### 【参数】

| 参数名称               | 描述                                                                    | 输入/输出 |
|--------------------|-----------------------------------------------------------------------|-------|
| pst_candi_corner   | 候选角点响应值图像指针。<br>不能为空。                                                 | 输入    |
| pst_corner         | 角点坐标信息指针。<br>不能为空。<br>内存至少需配置：<br>sizeof(xmedia_ive_st_corner_info_s) | 输出    |
| pst_st_corner_ctrl | 控制参数指针。<br>不能为空。                                                      | 输入    |

| 参数名称             | 支持图像类型 | 地址对齐    | 分辨率            |
|------------------|--------|---------|----------------|
| pst_candi_corner | U8C1   | 见【芯片差异】 | 16x16~1280x720 |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异          |
|------------|-------------|
| XM7606V13  | 地址为 16 字节对齐 |
| XM7206V11A | 地址为 8 字节对齐  |

#### 【需求】

- 头文件: xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件: libxmedia\_ive.a

#### 【注意】

- 与 OpenCV 中 ShiTomas 角点计算原理类似。
- pst\_corner→u16\_corner\_num 表示最终得到的角点数目。
- 使用该接口前必须调用 [xmedia\\_ive\\_st\\_candi\\_corner](#)，在保证 [xmedia\\_ive\\_st\\_candi\\_corner](#) 任务完成的情况下，使用 [xmedia\\_ive\\_st\\_candi\\_corner](#) 的输出 pst\_candi\_corner 作为该接口的参数输入。

#### 【举例】

无。

#### 【相关主题】

[xmedia\\_ive\\_st\\_candi\\_corner](#)

## xmedia\_ive\_sad

#### 【描述】

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】

```
xmedia_s32 xmedia_ive_sad(xmedia_ive_handle *p_xmedia_ive_handle, xmedia_ive_src_image_s *pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_sad, xmedia_ive_dst_image_s *pst_thr, xmedia_ive_sad_ctrl_s *pst_sad_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                  | 输入/输出 |
|---------------------|---------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。 | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。  | 输入    |

| 参数名称         | 描述                                                                                                                                             | 输入/输出 |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| pst_src2     | 源图像 2 指针。<br>不能为空。<br>高、宽同 pst_src1。                                                                                                           | 输入    |
| pst_sad      | 输出 SAD 图像指针。<br>根据 pst_sad_ctrl→en_out_ctrl, 若需要输出则不能为空。<br>根据 pst_sad_ctrl→en_mode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pst_src1 的 1/4、1/8、1/16。    | 输出    |
| pst_thr      | 输出 SAD 阈值化图像指针。<br>根据 pst_sad_ctrl→en_out_ctrl, 若需要输出则不能为空。<br>根据 pst_sad_ctrl→en_mode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pst_src1 的 1/4、1/8、1/16。 | 输出    |
| pst_sad_ctrl | 控制信息指针。<br>不能为空。                                                                                                                               | 输入    |
| b_instant    | 及时返回结果标志。                                                                                                                                      | 输入    |

| 参数名称     | 支持图像类型     | 地址对齐    | 分辨率                                                                             |
|----------|------------|---------|---------------------------------------------------------------------------------|
| pst_src1 | U8C1       | 1 byte  | 32x32 ~ 1920x1080                                                               |
| pst_src2 | U8C1       | 1 byte  | 同 pst_src1                                                                      |
| pst_sad  | U8C1、U16C1 | 见【芯片差异】 | 根据 pst_sad_ctrl→en_mode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pst_src1 的 1/4、1/8、1/16。 |

| 参数名称    | 支持图像类型 | 地址对齐    | 分辨率                                                                             |
|---------|--------|---------|---------------------------------------------------------------------------------|
| pst_thr | U8C1   | 见【芯片差异】 | 根据 pst_sad_ctrl→en_mode, 对应 4x4、8x8、16x16 分块模式, 高、宽分别为 pst_src1 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。 |

#### 【芯片差异】

| 芯片类型       | 差异          |
|------------|-------------|
| XM7606V13  | 地址为 16 字节对齐 |
| XM7206V11A | 地址为 8 字节对齐  |

#### 【需求】

- 头文件: xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件: libxmedia\_ive.a

#### 【注意】

计算公式如下:

$$SAD_{out}(x, y) = \sum_{\substack{n*x \leq i < n*(x+1) \\ n*y \leq j < n*(y+1)}} |I_1(i, j) - I_2(i, j)| ;$$

$$THR_{out}(x, y) = \begin{cases} minVal & (SAD_{out}(x, y) \leq Thr) \\ maxVal & (SAD_{out}(x, y) > Thr) \end{cases}$$

其中,  $I_1(i, j)$  对应 `pst_src1`,  $I_2(i, j)$  对应 `pst_src2`,  $SAD_{out}(x, y)$  对应 `pst_sad`,  $n$  与 `pst_sad_ctrl→en_mode` 相关, 对应 `XMEDIA_IVE_SAD_MODE_MB_4X4`、`XMEDIA_IVE_SAD_MODE_MB_8X8`、`XMEDIA_IVE_SAD_MODE_MB_16X16` 时分别取 4、8、16;

$THR_{out}(x, y)$  对应 `pst_thr`,  $Thr$ 、 $minVal$  和  $maxVal$  分别对应 `pst_sad_ctrl→u16_thr`、`pst_sad_ctrl→u8_min_val` 和 `pst_sad_ctrl→u8_max_val`。

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_resize\_csc\_letterbox

#### 【描述】

创建图像缩放、色域转换、加边框复合任务, 支持双线性缩放, 支持 yuv2rgb 色彩空间转换。

#### 【语法】

```
xmedia_s32 xmedia_ive_resize_csc_letterbox(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_resize_csc_letterbox_ctrl_s *pst_rcd_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                             | 描述                                               | 输入/输出 |
|----------------------------------|--------------------------------------------------|-------|
| <code>p_xmedia_ive_handle</code> | handle 指针。<br>不能为空。                              | 输出    |
| <code>pst_src</code>             | 源图像 1 指针。<br>不能为空。                               | 输入    |
| <code>pst_dst</code>             | 输出图像指针。<br>不能为空。<br>高、宽同 <code>pst_src1</code> 。 | 输出    |

| 参数名称         | 描述               | 输入/输出 |
|--------------|------------------|-------|
| pst_rcl_ctrl | 控制信息指针。<br>不能为空。 | 输入    |
| b_instant    | 及时返回结果标志。        | 输入    |

| 参数名称    | 支持图像类型                                             | 地址对齐    | 分辨率             |
|---------|----------------------------------------------------|---------|-----------------|
| pst_src | U8C1、<br>U8C3_PLANAR<br>(RGB/YUV444) 、<br>YUV420SP | 16 byte | 4x4 ~ 1920x1080 |
| pst_dst | U8C1、<br>U8C3_PLANAR(RGB)                          | 16 byte | 同 pst_src       |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 支持  |
| XM7206V11A | 不支持 |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无

### xmedia\_ive\_grad\_fg

#### 【描述】

根据背景图像和当前帧图像的梯度信息计算梯度前景图像。

#### 【语法】

```
xmedia_s32 xmedia_ive_grad_fg(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_bg_diff_fg, xmedia_ive_src_image_s *pst_cur_grad,
xmedia_ive_src_image_s *pst_bg_grad, xmedia_ive_dst_image_s *pst_grad_fg,
xmedia_ive_grad_fg_ctrl_s *pst_grad_fg_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                        | 输入/输出 |
|---------------------|-------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                       | 输出    |
| pst_bg_diff_fg      | 背景差分前景图像指针。<br>不能为空。                      | 输入    |
| pst_cur_grad        | 当前帧梯度图像指针。<br>不能为空。<br>高、宽同 pst_cur_grad。 | 输入    |
| pst_bg_grad         | 背景梯度图像指针。<br>不能为空。<br>高、宽同 pst_cur_grad。  | 输入    |
| pst_grad_fg         | 梯度前景图像指针。<br>不能为空。<br>高、宽同 pst_cur_grad。  | 输出    |
| pst_grad_fgCtrl     | 控制参数指针。<br>不能为空。                          | 输入    |
| b_instant           | 及时返回结果标志。                                 | 输入    |

| 参数名称         | 支持图像类型       | 地址对齐    | 分辨率               |
|--------------|--------------|---------|-------------------|
| pst_diff_fg  | S8C1         | 16 byte | 32x32 ~ 1920x1080 |
| pst_cur_grad | S8C2_PACKAGE | 16 byte | 同 pst_diff_fg     |
| pst_bg_grad  | S8C2_PACKAGE | 16 byte | 同 pstDiffFg       |
| pst_grad_fg  | U8C1         | 16 byte | 同 pstDiffFg       |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 支持  |
| XM7206V11A | 不支持 |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

背景梯度图像和当前梯度图像的类型为 S8C2\_PACKAGE，水平和竖直方向梯度按照 [xyxyxy...] 格式存储。

#### 【举例】

无。

#### 【相关主题】

- [xmedia\\_ive\\_match\\_bg\\_model](#)

- [xmedia\\_ive\\_update\\_bg\\_model](#)
- [xmedia\\_ive\\_gmm](#)

## xmedia\_ive\_match\_bg\_model

### 【描述】

基于 Codebook 演进的背景模型匹配。

### 【语法】

```
xmedia_s32 xmedia_ive_match_bg_model(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_src_image_s *pst_cur_img, xmedia_ive_data_s *pst_bg_model, xmedia_ive_image_s
*pst_fg_flag, xmedia_ive_dst_image_s *pst_bg_diff_fg, xmedia_ive_dst_image_s *pst_frm_diff_fg,
xmedia_ive_dst_mem_info_s *pst_stat_data, xmedia_ive_match_bg_model_ctrl_s
*pst_match_bg_model_ctrl, xmedia_bool b_instant);
```

### 【参数】

| 参数名称                | 描述                                                                                                                            | 输入/输出 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                                                                                                           | 输出    |
| pst_cur_img         | 当前帧灰度图像指针。<br>不能为空。                                                                                                           | 输入    |
| pst_bg_model        | 背景模型数据指针。<br>不能为空。<br>高同 pst_cur_img, 宽 =<br>pst_cur_img->u32_width *<br>sizeof( <a href="#">xmedia_ive_bg_model_pix_s</a> )。 | 输入、输出 |
| pst_fg_flag         | 前景状态图像指针。<br>不能为空。<br>高、宽同 pst_cur_img。                                                                                       | 输入、输出 |
| pst_bg_diff_fg      | 背景差分前景图像指针。<br>不能为空。<br>高、宽同 pst_cur_img。                                                                                     | 输出    |

| 参数名称                    | 描述                                                                  | 输入/输出 |
|-------------------------|---------------------------------------------------------------------|-------|
| pst_frm_diff_fg         | 帧间差分前景图像指针;<br>不能为空。<br>高、宽同 pst_cur_img。                           | 输出    |
| pst_stat_data           | 前景状态数据指针。<br>不能为空。<br>内存至少需配置<br>sizeof(xmedia_ive_fg_stat_data_s)。 | 输出    |
| pst_match_bg_model_ctrl | 控制参数指针。<br>不能为空。                                                    | 输入    |
| b_instant               | 及时返回结果标志。                                                           | 输入    |

| 参数名称            | 支持图像类型 | 地址对齐    | 分辨率               |
|-----------------|--------|---------|-------------------|
| pst_cur_img     | U8C1   | 16 byte | 32x32 ~ 1920x1080 |
| pst_bg_model    | -      | 16 byte | -                 |
| pst_fg_flag     | U8C1   | 16 byte | 同 pst_cur_img     |
| pst_bg_diff_fg  | S8C1   | 16 byte | 同 pst_cur_img     |
| pst_frm_diff_fg | S8C1   | 16 byte | 同 pst_cur_img     |
| pst_stat_data   | -      | 16 byte | -                 |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 支持  |
| XM7206V11A | 不支持 |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

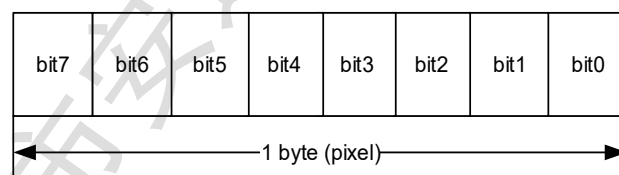
- 要求 pst\_fg\_flag、pst\_bg\_diff\_fg、pst\_frm\_diff\_fg 跨度一致。
- 背景模型数据 pst\_bg\_model 中每个像素以 xmedia\_ive\_bg\_model\_pix\_s (24 字节)表示, 即  

$$\text{pst\_model} \rightarrow \text{u16\_width} = \text{sizeof}(\text{xmedia\_ive\_bg\_model\_pix\_s}) * \text{pst\_src} \rightarrow \text{u16\_width},$$

$$\text{pst\_model} \rightarrow \text{u16\_height} = \text{pst\_src} \rightarrow \text{u16\_height},$$
至少需要分配内存大小为  

$$\text{lveGetStride}(\text{sizeof}(\text{xmedia\_ive\_bg\_model\_pix\_s}) * \text{pst\_src} \rightarrow \text{u16\_width}, \text{XMEDIA\_IVE\_STRIDE\_ALIGN}) * \text{pst\_model} \rightarrow \text{u16\_height}.$$
- 前景状态图像 pst\_fg\_flag 为 U8C1 类型, 其各比特位表示不同的状态信息, 单个像素各比特位示意图如图 2-17, 按从右到左由低位到高位顺序排布:

图2-17 前景状态标志图形单个像素各比特位示意图



其各个比特位表示的含义如下:

- 比特位只用到 bit0、bit1、bit2、bit5、bit6; 其中 bit0、bit1、bit2 是由本算子计算作为输出, bit5、bit6 是由外部函数计算作为输入。
- bit1 为 1 时表示像素为前景;
- bit1 为 1 且 bit0 为 1 时表示像素为运动前景;
- bit1 为 1 且 bit0 为 0 时表示像素为变化前景;
- bit2 为 1 时表示像素的背景模型处于工作状态;

- bit5 和 bit6 表示外部函数对前景状态的反馈，bit5 为 1 时表示前景像素需要短时间保持，bit6 为 1 时表示前景像素需要长时间保持。

#### 【举例】

无。

#### 【相关主题】

- [xmedia\\_ive\\_update\\_bg\\_model](#)
- [xmedia\\_ive\\_grad\\_fg](#)
- [xmedia\\_ive\\_gmm](#)

## xmedia\_ive\_update\_bg\_model

#### 【描述】

基于 Codebook 演进的背景模型更新，对背景模型的内部状态进行更新。

#### 【语法】

```
xmedia_s32 xmedia_ive_update_bg_model(xmedia_ive_handle *p_xmedia_ive_handle,
xmedia_ive_data_s *pst_bg_model, xmedia_ive_image_s *pst_fg_flag, xmedia_ive_dst_image_s
*pst_bg_img, xmedia_ive_dst_image_s *pst_chg_sta_img, xmedia_ive_dst_image_s
*pst_chg_sta_fg, xmedia_ive_dst_image_s *pst_chg_sta_life, xmedia_ive_dst_mem_info_s
*pst_stat_data, xmedia_ive_update_bg_model_ctrl_s *pst_update_bg_model_ctrl, xmedia_bool
b_instant);
```

#### 【参数】

| 参数名称                | 描述                                      | 输入/输出 |
|---------------------|-----------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                     | 输出    |
| pst_bg_model        | 背景模型数据指针。<br>不能为空。                      | 输入/输出 |
| pst_fg_flag         | 前景状态图像指针。<br>不能为空。                      | 输入/输出 |
| pst_bg_img          | 背景灰度图像指针。<br>不能为空。<br>高、宽同 pst_fg_flag。 | 输出    |

| 参数名称                     | 描述                                                                                                     | 输入/输出 |
|--------------------------|--------------------------------------------------------------------------------------------------------|-------|
| pst_chg_sta_img          | 变化状态灰度图像指针。<br>当 pst_update_bg_model_ctrl→<br>u8_det_chg_region 为 0 时, 可以为空。<br>高、宽同 pst_fg_flag。      | 输出    |
| pst_chg_sta_fg           | 变化状态前景图像指针。<br>当 pst_update_bg_model_ctrl→<br>u8_det_chg_region 为 0 时, 可以为空。<br>高、宽同 pst_fg_flag。      | 输出    |
| pst_chg_sta_life         | 变化状态像素的生命时间图像指针。<br>当 pst_update_bg_model_ctrl→<br>u8_det_chg_region 为 0 时, 可以为空。<br>高、宽同 pst_fg_flag。 | 输出    |
| pst_stat_data            | 背景状态数据指针。<br>不能为空。<br>内存至少需配置<br>sizeof(xmedia_ive_bg_stat_data_s)。                                    | 输出    |
| pst_update_bg_model_ctrl | 控制参数指针。<br>不能为空。                                                                                       | 输入    |
| b_instant                | 及时返回结果标志。                                                                                              | 输入    |

| 参数名称             | 支持图像类型 | 地址对齐    | 分辨率             |
|------------------|--------|---------|-----------------|
| pst_bg_model     | -      | 16 byte | -               |
| pst_fg_flag      | U8C1   | 16 byte | 32x32~1920*1080 |
| pst_bg_img       | U8C1   | 16 byte | 同 pst_fg_flag   |
| pst_chg_sta_img  | U8C1   | 16 byte | 同 pst_fg_flag   |
| pst_chg_sta_fg   | S8C1   | 16 byte | 同 pst_fg_flag   |
| pst_chg_sta_life | U16C1  | 16 byte | 同 pst_fg_flag   |

| 参数名称          | 支持图像类型 | 地址对齐    | 分辨率 |
|---------------|--------|---------|-----|
| pst_stat_data | -      | 16 byte | -   |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 支持  |
| XM7206V11A | 不支持 |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

- 要求 pst\_fg\_flag、pst\_bg\_img、pst\_chg\_sta\_img（不为空时）、pst\_chg\_sta\_fg（不为空时）跨度一致。
- 背景模型数据 pst\_model 参考 [xmedia\\_ive\\_match\\_bg\\_model](#) 中的说明。
- pst\_chg\_sta\_fg 表示变化状态前景图像，其中像素非 0 表示前景，否则表示背景。
- pst\_chg\_sta\_life 表示变化状态前景像素的生命时间图像，其像素值表示变化前景的持续时间。
- 变化状态指像素值发生变化而成为前景，并且变化后的像素值较长时间都保持稳定的状态，这一般是由静止遗留物或者静止移走物在图像中产生。

#### 【举例】

无。

#### 【相关主题】

- [xmedia\\_ive\\_match\\_bg\\_model](#)
- [xmedia\\_ive\\_grad\\_fg](#)
- [xmedia\\_ive\\_gmm](#)

## xmedia\_ive\_lbp

#### 【描述】

创建 LBP 计算任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_lbp(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_lbp_ctrl_s *pst_lbp_ctrl, xmedia_bool
b_instant);
```

#### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入    |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_lbp_ctrl        | 控制信息指针。<br>不能为空。                  | 输入    |
| b_instant           | 及时返回结果标志。                         | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率             |
|---------|--------|--------|-----------------|
| pst_src | U8C1   | 8 byte | 32x32~1920x1024 |
| pst_dst | U8C1   | 8 byte | 32x32~1920x1024 |

【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

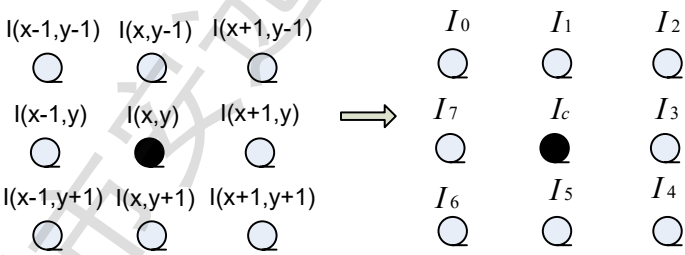
【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

【注意】

LBP 计算公式如图 2-18 所示。

图2-18 LBP 计算公式示意图



- XMEDIA\_IVE\_LBP\_CMP\_NORMAL

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

- XMEDIA\_IVE\_LBP\_CMP\_ABS

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

- 其中,  $I(x, y)$  对应 `pst_src`,  $lpb(x, y)$  对应 `pst_dst`,  $thr$  对应 `pst_lbp_ctrl` → `un8_bit_thr`。

#### 【举例】

无。

#### 【相关主题】

无。

## xmedia\_ive\_acc

#### 【描述】

创建两图像像素累加运算任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_acc(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_acc_ctrl_s *pst_acc_ctrl, xmedia_bool
b_instant);
```

#### 【参数】

| 参数名称                             | 描述                                           | 输入/输出 |
|----------------------------------|----------------------------------------------|-------|
| <code>p_xmedia_ive_handle</code> | handle 指针。<br>不能为空。                          | 输出    |
| <code>pst_src</code>             | 源图像指针。<br>不能为空。                              | 输入    |
| <code>pst_dst</code>             | 输出图像指针。<br>高、宽同 <code>pst_src</code> ; 不能为空。 | 输出    |
| <code>pst_acc_ctrl</code>        | 控制参数指针。<br>不能为空。                             | 输入    |

| 参数名称                 | 支持图像类型    | 地址对齐   | 分辨率               |
|----------------------|-----------|--------|-------------------|
| <code>pst_src</code> | U8C1 的二值图 | 8 byte | 32x32 ~ 1920x1080 |

| 参数名称    | 支持图像类型          | 地址对齐   | 分辨率       |
|---------|-----------------|--------|-----------|
| pst_dst | U32C1、<br>U64C1 | 8 byte | 同 pst_src |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

xmedia\_ive\_lk\_optical\_flow\_pyr

#### 【描述】

创建多层 LK 光流计算任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_lk_optical_flow_pyr(xmedia_ive_handle
*p_ive_handle,xmedia_ive_src_image_s ast_src_prev_pyr[],xmedia_ive_src_image_s
ast_src_nextpyr[], xmedia_ive_src_mem_info_s *pst_prev_pts,xmedia_ive_mem_info_s
```

\*pst\_next\_pts, xmedia\_ive\_dst\_mem\_info\_s \*pst\_status, xmedia\_ive\_dst\_mem\_info\_s  
 \*pst\_err, xmedia\_ive\_lk\_optical\_flow\_pyr\_ctrl\_s \*pst\_lk\_opti\_flow\_pyr\_ctrl, xmedia\_bool b\_instant);

#### 【参数】

| 参数名称                | 描述                                                                                                                                                                                                                                                                                          | 输入/输出 |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                                                                                                                                                                                                                                                                         | 输出    |
| ast_src_prev_pyr    | 前一帧图像的金字塔图像数组，金字塔层数由<br>pst_lk_opti_flow_pyr_ctrl→u8_max_level 控制。<br>不能为空。                                                                                                                                                                                                                 | 输入    |
| ast_src_nextpyr     | 下一帧图像的金字塔图像数组，与<br>ast_src_prev_pyr []有相同的层数，每层图像<br>大小类型均相同。<br>不能为空。                                                                                                                                                                                                                      | 输入    |
| pst_prev_pts        | 前一帧图像金字塔第 0 层 (ast_src_prev_pyr<br>[0]) 的初始特征点数组。<br>不能为空。坐标只能为<br>xmedia_ive_point_s25q7_s 类型；内存至少需<br>分配：<br>pst_lk_opti_flow_pyr_ctrl→u16_pts_num *<br>sizeof(xmedia_ive_point_s25q7_s)。                                                                                                 | 输入    |
| pst_next_pts        | 特征点 pst_prev_pts 经过金字塔计算 LK 光流<br>得到对应于下一帧图像金字塔第 0 层<br>(astSrcNextPyr[]) 的坐标。当<br>pst_lk_opti_flow_pyr_ctrl→b_use_init_flow 为<br>true 时，需要初始化该特征点数组。不能为<br>空。坐标只能为 xmedia_ive_point_s25q7_s<br>类型；内存至少需分配：<br>pst_lk_opti_flow_pyr_ctrl→u16_pts_num *<br>sizeof(xmedia_ive_point_s25q7_s)。 | 输入、输出 |
| pst_status          | pst_next_pts 中每个特征点对应一个<br>xmedia_u8 的跟踪状态信息，1 表示成功，0<br>表示失败。                                                                                                                                                                                                                              | 输入、输出 |

| 参数名称                  | 描述                                                                                         | 输入/输出 |
|-----------------------|--------------------------------------------------------------------------------------------|-------|
| pst_err               | 对 pst_next_pts 中每个跟踪成功的特征点，对比 pst_prev_pts 中对应特征点周边进行的相似度误差估计（xmedia_u9q7 类型），跟踪失败的特征点不做估计 | 输入、输出 |
| pst_lk_opti_flow_ctrl | 控制参数指针。<br>不能为空。                                                                           | 输入    |
| b_instant             | 及时返回结果标志。                                                                                  | 输入    |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【注意】

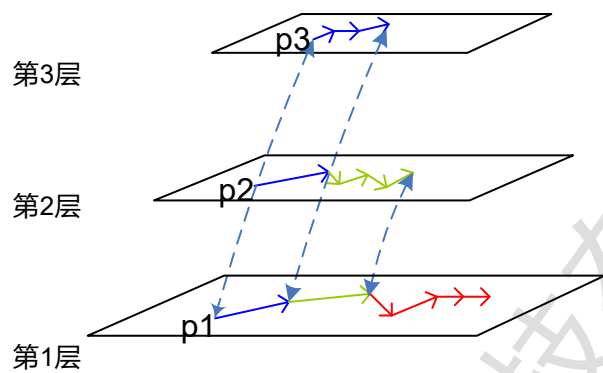
- 求解下面的光流方程中，仅用到特征点周围 7X7 像素的来计算对应的  $I_x$ 、 $I_y$ 、 $I_t$

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

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 2-19 所示。

图2-19 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： $p_0$ ， $p_1$ ， $p_2$ ；
- 以  $p_2$  和初始为 0 的  $mv_2$  作为输入调用 LK 算子求出在第 2 层上的位移  $mv_2$ ；
- 以  $p_1$  和  $mv_2$  作为输入调用 LK 算子求出第 1 层上的位移  $mv_1$ ；
- 以  $p_0$  和  $mv_1$  作为输入调用 LK 算子求出第 0 层上的位移  $mv_0$ ；
- 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。

xmedia\_ive\_and\_ccl

【描述】

创建两二值图像相与&连通区域标记组合任务。

### 【语法】

```
xmedia_s32 xmedia_ive_and_ccl(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst, xmedia_ive_dst_mem_info_s *pst_blob, xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl, xmedia_bool
b_instant);
```

### 【参数】

| 参数名称                | 描述                                                                                     | 输入/输出 |
|---------------------|----------------------------------------------------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                                                                    | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。                                                                     | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。                                                                     | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。                                                         | 输出    |
| pst_blob            | 连通区域信息指针。<br>不能为空。<br>内存至少需配置为 sizeof<br>(xmedia_ive_ccblob_s)大小，最多输出 254<br>个有效的连通区域。 | 输出    |
| pst_ccl_ctrl        | 控制参数指针<br>不能为空。                                                                        | 输入    |
| b_instant           | 及时返回结果标志。                                                                              | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率              |
|----------|--------|-------|------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1280x720 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1       |
| pst_dst  | U8C1   | 8byte | 同 pst_src1       |

### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

### 【举例】

无。

### 【相关主题】

无。

xmedia\_ive\_or\_ccl

### 【描述】

创建两二值图像相或&连通区域标记组合任务。

### 【语法】

```
xmedia_s32 xmedia_ive_or_ccl(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst, xmedia_ive_dst_mem_info_s *pst_blob, xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl, xmedia_bool
b_instant);
```

### 【参数】

| 参数名称                | 描述                                                                               | 输入/输出 |
|---------------------|----------------------------------------------------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                                                              | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。                                                               | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。                                                               | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。                                                   | 输出    |
| pst_blob            | 连通区域信息指针。<br>不能为空。<br>内存至少需配置为 sizeof (xmedia_ive_ccblob_s)大小，最多输出 254 个有效的连通区域。 | 输出    |
| pst_ccl_ctrl        | 控制参数指针<br>不能为空。                                                                  | 输入    |
| b_instant           | 及时返回结果标志。                                                                        | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率              |
|----------|--------|-------|------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1280x720 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1       |
| pst_dst  | U8C1   | 8byte | 同 pst_src1       |

#### 【返回值】

| 返回值 | 描述  |
|-----|-----|
| 0   | 成功。 |

| 返回值 | 描述        |
|-----|-----------|
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_xor\_ccl

#### 【描述】

创建两二值图的异或&连通区域标记组合任务。

#### 【语法】

```
xmedia_ive_xor_ccl(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1,xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_dst_mem_info_s *pst_blob,xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl, xmedia_bool
b_instant);
```

#### 【参数】

| 参数名称                | 描述                  | 输入/输出 |
|---------------------|---------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。 | 输出    |

| 参数名称         | 描述                                                                               | 输入/输出 |
|--------------|----------------------------------------------------------------------------------|-------|
| pst_src1     | 源图像 1 指针。<br>不能为空。                                                               | 输入、输出 |
| pst_src2     | 源图像 2 指针。<br>不能为空。                                                               | 输出    |
| pst_dst      | 输出图像指针，高、宽同 pst_src1。<br>不能为空。                                                   | 输出    |
| pst_blob     | 连通区域信息指针。<br>不能为空。<br>内存至少需配置为 sizeof (xmedia_ive_ccblob_s)大小，最多输出 254 个有效的连通区域。 | 输出    |
| pst_ccl_ctrl | 控制参数指针<br>不能为空。                                                                  | 输入    |
| b_instant    | 及时返回结果标志。                                                                        | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率              |
|----------|--------|-------|------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1280x720 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1       |
| pst_dst  | U8C1   | 8byte | 同 pst_src1       |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_sub\_ccl

#### 【描述】

创建两灰度图像相减&连通区域标记组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_sub_ccl(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst, xmedia_ive_dst_mem_info_s *pst_blob, xmedia_ive_sub_ctrl_s
*pst_sub_ctrl, xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                  | 输入/输出 |
|---------------------|---------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。 | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。  | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。  | 输出    |

| 参数名称         | 描述                                                                               | 输入/输出 |
|--------------|----------------------------------------------------------------------------------|-------|
| pst_dst      | 输出图像指针，高、宽同 pst_src1。<br>不能为空。                                                   | 输出    |
| pst_blob     | 连通区域信息指针。<br>不能为空。<br>内存至少需配置为 sizeof (xmedia_ive_ccblob_s)大小，最多输出 254 个有效的连通区域。 | 输出    |
| pst_sub_ctrl | 控制信息指针。<br>不能为空。                                                                 | 输入    |
| pst_ccl_ctrl | 控制参数指针<br>不能为空。                                                                  | 输入    |
| b_instant    | 及时返回结果标志。                                                                        | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率              |
|----------|--------|-------|------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1280x720 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1       |
| pst_dst  | U8C1   | 8byte | 同 pst_src1       |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型      | 差异  |
|-----------|-----|
| XM7606V13 | 不支持 |

|            |    |
|------------|----|
| XM7206V11A | 支持 |
|------------|----|

【需求】

- 头文件: xmedia\_ive\_common.h、 xmedia\_ive.h
- 库文件: libxmedia\_ive.a

【举例】

无。

【相关主题】

无。

xmedia\_ive\_add\_ccl

【描述】

创建两二值图像相与&连通区域标记组合任务。

【语法】

```
xmedia_s32 xmedia_ive_add_ccl(xmedia_ive_handle *p_ive_handle,xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst,xmedia_ive_dst_mem_info_s *pst_blob, xmedia_ive_add_ctrl_s
*pst_add_ctr,xmedia_ive_ccl_ctrl_s *pst_ccl_ctrl, xmedia_bool b_instant);
```

【参数】

| 参数名称                | 描述                             | 输入/输出 |
|---------------------|--------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。            | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。             | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。             | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。 | 输出    |

| 参数名称         | 描述                                                                               | 输入/输出 |
|--------------|----------------------------------------------------------------------------------|-------|
| pst_blob     | 连通区域信息指针。<br>不能为空。<br>内存至少需配置为 sizeof (xmedia_ive_ccblob_s)大小，最多输出 254 个有效的连通区域。 | 输出    |
| pst_add_ctrl | 控制参数指针。<br>不能为空。                                                                 | 输入    |
| pst_ccl_ctrl | 控制参数指针<br>不能为空。                                                                  | 输入    |
| b_instant    | 及时返回结果标志。                                                                        | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率              |
|----------|--------|-------|------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1280x720 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1       |
| pst_dst  | U8C1   | 8byte | 同 pst_src1       |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

### 【需求】

- 头文件: xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件: libxmedia\_ive.a

### 【举例】

无。

### 【相关主题】

无。

## xmedia\_ive\_sub\_hist

### 【描述】

创建两灰度图像相减&直方图统计组合任务。

### 【语法】

```
xmedia_s32 xmedia_ive_sub_hist(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_mem_info_s *pst_dst,
xmedia_ive_sub_ctrl_s *pst_sub_ctrl, xmedia_bool b_instant);
```

### 【参数】

| 参数名称                | 描述                             | 输入/输出 |
|---------------------|--------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。            | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。             | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。             | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。 | 输出    |
| pst_sub_ctrl        | 控制参数指针。<br>不能为空。               | 输入    |
| b_instant           | 及时返回结果标志。                      | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率               |
|----------|--------|-------|-------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

xmedia\_ive\_sub\_thresh

#### 【描述】

创建两灰度图像相减&阈值化组合任务。

### 【语法】

```
xmedia_s32 xmedia_ive_sub_thresh(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1,xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s
*pst_dst,xmedia_ive_sub_ctrl_s *pst_sub_ctrl,xmedia_ive_thresh_ctrl_s *pst_thr_ctrl,
xmedia_bool b_instant);
```

### 【参数】

| 参数名称                | 描述                             | 输入/输出 |
|---------------------|--------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。            | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。             | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。             | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。 | 输出    |
| pst_sub_ctrl        | 控制参数指针。<br>不能为空。               | 输入    |
| pst_thr_ctrl        | 控制信息指针。<br>不能为空。               | 输入    |
| b_instant           | 及时返回结果标志。                      | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率               |
|----------|--------|-------|-------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8byte | 同 pst_src1        |

### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_sub\_integ

#### 【描述】

创建两灰度图像相减&积分图计算组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_sub_integ(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_sub_ctrl_s *pst_sub_ctrl, xmedia_ive_integ_ctrl_s *pst_integ_ctrl,
xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                             | 输入/输出 |
|---------------------|--------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。            | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。             | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。             | 输出    |
| pst_dst             | 输出图像指针，高、宽同 pst_src1。<br>不能为空。 | 输出    |
| pst_sub_ctrl        | 控制参数指针。<br>不能为空。               | 输入    |
| pst_integ_ctrl      | 控制信息指针。<br>不能为空。               | 输入    |
| b_instant           | 及时返回结果标志。                      | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率               |
|----------|--------|-------|-------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_sub\_map

#### 【描述】

创建两灰度图像相减&map（映射赋值）组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_sub_map(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_src_mem_info_s *pst_map,
xmedia_ive_dst_image_s *pst_dst, xmedia_ive_sub_ctrl_s *pst_sub_ctrl,
xmedia_ive_map_ctrl_s *pst_map_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                  | 输入/输出 |
|---------------------|---------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。 | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。  | 输入、输出 |
| pst_src2            | 源图像 2 指针。<br>不能为空。  | 输出    |

| 参数名称      | 描述                             | 输入/输出 |
|-----------|--------------------------------|-------|
| pst_map   | 映射表信息指针。<br>不能为空。              | 输入    |
| pst_dst   | 输出图像指针，高、宽同 pst_src1。<br>不能为空。 | 输出    |
| b_instant | 及时返回结果标志。                      | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐  | 分辨率               |
|----------|--------|-------|-------------------|
| pst_src1 | U8C1   | 8byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

【相关主题】

无。

## xmedia\_ive\_add\_hist

【描述】

创建两二值图像加权加&直方图统计组合任务。

【语法】

```
xmedia_s32 xmedia_ive_add_hist(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_mem_info_s *pst_dst,
xmedia_ive_add_ctrl_s *pst_add_ctr, xmedia_bool b_instant);
```

【参数】

| 参数名称                | 描述                                   | 输入/输出 |
|---------------------|--------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                  | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。                   | 输入    |
| pst_src2            | 源图像 2 指针。<br>不能为空。<br>高、宽同 pst_src1。 | 输入    |
| pst_dst             | 输出图像指针。<br>高、宽同 pst_src1; 不能为空。      | 输出    |
| pst_add_ctrl        | 控制参数指针。<br>不能为空。                     | 输入    |
| b_instant           | 及时返回结果标志。                            | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐   | 分辨率               |
|----------|--------|--------|-------------------|
| pst_src1 | U8C1   | 8 byte | 32x32 ~ 1920x1080 |

| 参数名称     | 支持图像类型 | 地址对齐   | 分辨率        |
|----------|--------|--------|------------|
| pst_src2 | U8C1   | 8 byte | 同 pst_src1 |
| pst_dst  | U8C1   | 8 byte | 同 pst_src1 |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

xmedia\_ive\_add\_thresh

#### 【描述】

创建两二值图像加权加&阈值化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_add_thresh(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst,
```

```
xmedia_ive_add_ctrl_s *pst_add_ctrl, xmedia_ive_thresh_ctrl_s *pst_thr_ctrl,
xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                   | 输入/输出 |
|---------------------|--------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                  | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。                   | 输入    |
| pst_src2            | 源图像 2 指针。<br>不能为空。<br>高、宽同 pst_src1。 | 输入    |
| pst_dst             | 输出图像指针。<br>高、宽同 pst_src1；不能为空。       | 输出    |
| pst_add_ctrl        | 控制参数指针。<br>不能为空。                     | 输入    |
| pst_thr_ctrl        | 控制信息指针。<br>不能为空。                     | 输入    |
| b_instant           | 及时返回结果标志。                            | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐   | 分辨率               |
|----------|--------|--------|-------------------|
| pst_src1 | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8 byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8 byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_add\_integ

#### 【描述】

创建两二值图像加权加&积分图计算组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_add_integ(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_dst_image_s *pst_dst,
xmedia_ive_add_ctrl_s *pst_add_ctr, xmedia_ive_integ_ctrl_s *pst_integ_ctrl, xmedia_bool
b_instant);
```

#### 【参数】

| 参数名称                | 描述                                   | 输入/输出 |
|---------------------|--------------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。                  | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。                   | 输入    |
| pst_src2            | 源图像 2 指针。<br>不能为空。<br>高、宽同 pst_src1。 | 输入    |
| pst_dst             | 输出图像指针。<br>高、宽同 pst_src1; 不能为空。      | 输出    |
| pst_add_ctrl        | 控制参数指针。<br>不能为空。                     | 输入    |
| pst_integ_ctrl      | 控制信息指针。<br>不能为空。                     | 输入    |
| b_instant           | 及时返回结果标志。                            | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐   | 分辨率               |
|----------|--------|--------|-------------------|
| pst_src1 | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8 byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8 byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述         |
|-----|------------|
| 0   | 成功。        |
| 非 0 | 失败, 参见错误码。 |

### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

### 【需求】

- 头文件: xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件: libxmedia\_ive.a

### 【举例】

无。

### 【相关主题】

无。

## xmedia\_ive\_add\_map

### 【描述】

创建两二值图像加权加&map（映射赋值）组合任务。

### 【语法】

```
xmedia_s32 xmedia_ive_add_map(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src1, xmedia_ive_src_image_s *pst_src2, xmedia_ive_src_mem_info_s *pst_map,
xmedia_ive_dst_image_s *pst_dst, xmedia_ive_add_ctrl_s *pst_add_ctr,
xmedia_ive_map_ctrl_s *pst_map_ctrl, xmedia_bool b_instant);
```

### 【参数】

| 参数名称                | 描述                  | 输入/输出 |
|---------------------|---------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。 | 输出    |
| pst_src1            | 源图像 1 指针。<br>不能为空。  | 输入    |

| 参数名称         | 描述                                   | 输入/输出 |
|--------------|--------------------------------------|-------|
| pst_src2     | 源图像 2 指针。<br>不能为空。<br>高、宽同 pst_src1。 | 输入    |
| pst_map      | 映射表信息指针。<br>不能为空。                    | 输入    |
| pst_dst      | 输出图像指针。<br>高、宽同 pst_src1；不能为空。       | 输出    |
| pst_add_ctrl | 控制参数指针。<br>不能为空。                     | 输入    |
| pst_map_ctrl | 控制参数指针。<br>不能为空。                     | 输入    |
| b_instant    | 及时返回结果标志。                            | 输入    |

| 参数名称     | 支持图像类型 | 地址对齐   | 分辨率               |
|----------|--------|--------|-------------------|
| pst_src1 | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_src2 | U8C1   | 8 byte | 同 pst_src1        |
| pst_dst  | U8C1   | 8 byte | 同 pst_src1        |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型 | 差异 |
|------|----|
|------|----|

|            |     |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件: xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件: libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_filter\_thresh

#### 【描述】

创建 5x5 模板滤波 & 阈值化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_filter_thresh(xmedia_ive_handle *p_ive_handle, xmedia_ive_src_image_s
*pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_filter_ctrl_s *pstflt_ctrl,
xmedia_ive_thresh_ctrl_s *pst_thr_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |

| 参数名称         | 描述              | 输入/输出 |
|--------------|-----------------|-------|
| pst_flt_ctrl | 控制参数指针<br>不能为空。 | 输入    |
| pst_thr_ctrl | 控制参数指针<br>不能为空。 | 输入    |
| b_instant    | 及时返回结果标志。       | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_ord\_stat\_filter\_thresh

#### 【描述】

创建 3\*3 模板顺序统计量滤波 (OrdstatFilter) & 阈值化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_ord_stat_filter_thresh (xmedia_ive_handle *p_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s
*pst_dst, xmedia_ive_ord_stat_filter_ctrl_s *pst_ord_statflt_ctrl, xmedia_ive_thresh_ctrl_s
*pst_thr_ctrl, xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                 | 描述                                | 输入/输出 |
|----------------------|-----------------------------------|-------|
| p_xmedia_ive_handle  | handle 指针。<br>不能为空。               | 输出    |
| pst_src              | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst              | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_ord_statflt_ctrl | 控制参数指针<br>不能为空。                   | 输入    |
| pst_thr_ctrl         | 控制参数指针<br>不能为空。                   | 输入    |
| b_instant            | 及时返回结果标志。                         | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率       |
|---------|--------|--------|-----------|
| pst_dst | U8C1   | 8 byte | 同 pst_src |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

## xmedia\_ive\_sobel\_thresh\_s16

#### 【描述】

创建 5x5 模板 sobel-like 梯度计算 & s16 数据到 8bit 数据的阈值化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_sobel_thresh_s16 (xmedia_ive_handle *p_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_sobel_ctrl_s
*pst_sobel_ctrl, xmedia_ive_thresh_s16_ctrl_s *pst_thr_s16_ctrl, xmedia_bool b_instant);
```

### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_sobel_ctrl      | 控制参数指针<br>不能为空。                   | 输入    |
| pst_thr_s16_ctrl    | 控制参数指针<br>不能为空。                   | 输入    |
| b_instant           | 及时返回结果标志。                         | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

### 【芯片差异】

| 芯片类型 | 差异 |
|------|----|
|------|----|

|            |     |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_mag\_and\_ang\_thresh\_u16

#### 【描述】

创建 5x5 模板梯度幅值& 幅角计算 & u16 数据到 u8 数据的阈值化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_mag_and_ang_thresh_u16 (xmedia_ive_handle *p_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s
*pst_dst,xmedia_ive_mag_and_ang_ctrl_s *pst_mag_and_ang_ctrl, xmedia_ive_thresh_u16_ctrl_s
*pst_thr_u16_ctrl,xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |

| 参数名称                 | 描述               | 输入/输出 |
|----------------------|------------------|-------|
| pst_mag_and_ang_ctrl | 控制参数指针。<br>不能为空。 | 输入    |
| pst_thr_u16_ctrl     | 控制参数指针。<br>不能为空。 | 输入    |
| b_instant            | 及时返回结果标志。        | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_mag\_and\_ang\_16bit\_to\_8bit

#### 【描述】

创建 5x5 模板梯度幅值& 幅角计算 & 16bit 图像数据到 8bit 图像数据的线性转化组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_mag_and_ang_16bit_to_8bit (xmedia_ive_handle *p_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s
*pst_dst,xmedia_ive_mag_and_ang_ctrl_s *pst_mag_and_ang_ctrl,
xmedia_ive_16bit_to_8bit_ctrl_s *pst_16bit_to_8bit_ctrl,xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                   | 描述                                | 输入/输出 |
|------------------------|-----------------------------------|-------|
| p_xmedia_ive_handle    | handle 指针。<br>不能为空。               | 输出    |
| pst_src                | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst                | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_mag_and_ang_ctrl   | 控制参数指针<br>不能为空。                   | 输入    |
| pst_16bit_to_8bit_ctrl | 控制参数指针<br>不能为空。                   | 输入    |
| b_instant              | 及时返回结果标志。                         | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

xmedia\_ive\_thresh\_dilate

#### 【描述】

创建灰度图像阈值化& 5x5 模板膨胀组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_thresh_dilate(xmedia_ive_handle *p_ive_handle,
xmedia_ive_src_image_s *pst_src, xmedia_ive_dst_image_s *pst_dst, xmedia_ive_thresh_ctrl_s
```

\*pst\_thr\_ctrl, xmedia\_ive\_dilate\_ctrl\_s \*pst\_dilate\_ctrl, xmedia\_bool b\_instant);

#### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |
| pst_thr_ctrl        | 控制参数指针。<br>不能为空。                  | 输入    |
| pst_dilate_ctrl     | 控制参数指。<br>不能为空。                   | 输入    |
| b_instant           | 及时返回结果标志。                         | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

#### 【相关主题】

无。

### xmedia\_ive\_thresh\_erode

#### 【描述】

创建灰度图像阈值化& 5x5 模板腐蚀组合任务。

#### 【语法】

```
xmedia_s32 xmedia_ive_thresh_erode(xmedia_ive_handle
*p_ive_handle,xmedia_ive_src_image_s *pst_src,xmedia_ive_dst_image_s
*pst_dst,xmedia_ive_thresh_ctrl_s *pst_thr_ctrl,xmedia_ive_erode_ctrl_s
*pst_erode_ctrl,xmedia_bool b_instant);
```

#### 【参数】

| 参数名称                | 描述                                | 输入/输出 |
|---------------------|-----------------------------------|-------|
| p_xmedia_ive_handle | handle 指针。<br>不能为空。               | 输出    |
| pst_src             | 源图像指针。<br>不能为空。                   | 输入、输出 |
| pst_dst             | 输出图像指针。<br>不能为空。<br>高、宽同 pst_src。 | 输出    |

| 参数名称           | 描述              | 输入/输出 |
|----------------|-----------------|-------|
| pst_thr_ctrl   | 控制参数指针<br>不能为空。 | 输入    |
| pst_erode_ctrl | 控制参数指针<br>不能为空。 | 输入    |
| b_instant      | 及时返回结果标志。       | 输入    |

| 参数名称    | 支持图像类型 | 地址对齐   | 分辨率               |
|---------|--------|--------|-------------------|
| pst_src | U8C1   | 8 byte | 32x32 ~ 1920x1080 |
| pst_dst | U8C1   | 8 byte | 同 pst_src         |

#### 【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

#### 【芯片差异】

| 芯片类型       | 差异  |
|------------|-----|
| XM7606V13  | 不支持 |
| XM7206V11A | 支持  |

#### 【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

#### 【举例】

无。

【相关主题】

无。

## xmedia\_ive\_query

【描述】

查询已创建任务完成情况。

【语法】

```
xmedia_s32 xmedia_ive_query(xmedia_ive_handle handle, xmedia_bool *p_b_finish, xmedia_bool b_block);
```

【参数】

| 参数名称       | 描述                 | 输入/输出 |
|------------|--------------------|-------|
| handle     | 任务的 handle。        | 输入    |
| p_b_finish | 任务完成状态指针。<br>不能为空。 | 输出    |
| b_block    | 是否阻塞查询标志。          | 输入    |

【返回值】

| 返回值 | 描述        |
|-----|-----------|
| 0   | 成功。       |
| 非 0 | 失败，参见错误码。 |

【需求】

- 头文件：xmedia\_ive\_common.h、xmedia\_ive.h
- 库文件：libxmedia\_ive.a

【注意】

- 在用户使用 IVE 任务结果前，为确保 IVE 任务已完成，用户可以使用阻塞方式调用此接口查询。

- IVE 内部是按任务创建顺序依次执行任务的，所以用户不必每次都使用查询接口，如用户依次创建了 A, B 两个任务，那么如果 B 任务完成了，这个时候 A 任务肯定也完成了，此时使用 A 任务的结果时不必再次调用查询接口。
- 返回值为 ERR\_CODE\_IVE\_QUERY\_TIMEOUT（查询超时）时，可以继续查询。
- 返回值为 ERR\_CODE\_IVE\_SYS\_TIMEOUT（系统超时）时，用户的 IVE 任务必须全部重新提交。

**【举例】**

无

**【相关主题】**

无

# 3

## 数据类型和数据结构

IVE 模块相关数据类型和数据结构定义如下：

- `XMEDIA_IVE_HIST_NUM`：定义直方图统计 bin 数目。
- `XMEDIA_IVE_MAP_NUM`：定义映射查找表项数目。
- `XMEDIA_IVE_MAX_REGION_NUM`：定义最大连通区域数目。
- `XMEDIA_IVE_ST_MAX_CORNER_NUM`：定义 Shi-Tomasi-like 角点最大数目。
- `xmedia_ive_image_type_e`：定义二维广义图像支持的图像类型。
- `xmedia_en_ive_err_code_e`：定义错误码。
- `xmedia_ive_image_s`：定义二维广义图像信息。
- `xmedia_ive_src_image_s`：定义源图像。
- `xmedia_ive_dst_image_s`：定义输出图像。
- `xmedia_ive_data_s`：定义以 byte 为单位的二维数据信息。
- `xmedia_ive_src_data_s`：定义以 byte 为单位的二维源数据信息。
- `xmedia_ive_dst_data_s`：定义 byte 为单位的二维输出数据信息。
- `xmedia_ive_mem_info_s`：定义一维数据内存信息。
- `xmedia_ive_src_mem_info_s`：定义一维源数据。
- `xmedia_ive_dst_mem_info_s`：定义一维输出数据。
- `xmedia_ive_8bit_u`：定义 8bit 数据联合体。
- `xmedia_ive_point_u16_s`：定义 U16 表示的点信息结构体。
- `xmedia_ive_point_s16_s`：定义 S16 表示的点信息结构体。
- `xmedia_ive_point_u14q2_s`：定义 U14Q2 坐标结构体。
- `xmedia_ive_point_s25q7_s`：定义 S25Q7 定点表示的点信息结构体。

- `xmedia_ive_rect_u16_s`: 定义 U16 表示的矩形信息结构体。
- `xmedia_ive_rect_u32_s`: 定义 U32 表示的矩形信息结构体。
- `xmedia_ive_rect_s24q8_s`: 定义 S24Q8 表示的矩形信息结构体。
- `xmedia_ive_dma_mode_e`: 定义 DMA 操作模式。
- `xmedia_ive_dma_ctrl_s`: 定义 DMA 控制信息。
- `xmedia_ive_filter_ctrl_s`: 定义模板滤波控制信息。
- `xmedia_ive_csc_mode_e`: 定义色彩空间转换模式。
- `xmedia_ive_csc_ctrl_s`: 定义色彩空间转换控制信息。
- `xmedia_ive_filter_and_csc_ctrl_s`: 定义模板滤波加色彩空间转换复合功能控制信息。
- `xmedia_ive_sobel_out_ctrl_e`: 定义 Sobel 输出控制信息。
- `xmedia_ive_sobel_ctrl_s`: 定义 Sobel-like 梯度计算控制信息。
- `xmedia_ive_mag_and_ang_out_ctrl_e`: 定义梯度幅值与角度计算的输出格式。
- `xmedia_ive_mag_and_ang_ctrl_s`: 定义梯度幅值和幅角计算的控制信息。
- `xmedia_ive_dilate_ctrl_s`: 定义膨胀控制信息。
- `xmedia_ive_erode_ctrl_s`: 定义腐蚀控制信息。
- `xmedia_ive_thresh_mode_e`: 定义图像二值化输出格式。
- `xmedia_ive_thresh_ctrl_s`: 定义图像二值化控制信息。
- `xmedia_ive_sub_mode_e`: 定义两图像相减输出格式。
- `xmedia_ive_sub_ctrl_s`: 定义两图像相减控制参数。
- `xmedia_ive_integ_out_ctrl_e`: 定义积分图输出控制参数。
- `xmedia_ive_integ_ctrl_s`: 定义积分图计算控制参数。
- `xmedia_ive_thresh_s16_mode_e`: 定义 16bit 有符号图像的阈值化模式。
- `xmedia_ive_thresh_s16_ctrl_s`: 定义 16bit 有符号图像的阈值化控制参数。
- `xmedia_ive_thresh_u16_mode_e`: 定义 16bit 无符号图像的阈值化模式。
- `xmedia_ive_thresh_u16_ctrl_s`: 定义 16bit 无符号图像的阈值化控制参数。
- `xmedia_ive_16bit_to_8bit_mode_e`: 定义 16bit 图像数据到 8bit 图像数据的转化模式。
- `xmedia_ive_16bit_to_8bit_ctrl_s`: 定义 16bit 图像数据到 8bit 图像数据的转化控制参数。
- `xmedia_ive_ord_stat_filter_mode_e`: 定义顺序统计量滤波模式。

- `xmedia_ive_ord_stat_filter_ctrl_s`: 定义顺序统计量滤波控制参数。
- `xmedia_ive_map_lut_mem_s`: 定义 Map 算子的查找表内存信息。
- `xmedia_ive_map_u8bit_lut_mem_s`: 定义 Map U8C1→U8C1 的查找表内存。
- `xmedia_ive_map_u16bit_lut_mem_s`: 定义 Map U8C1→U16C1 的查找表内存。
- `xmedia_ive_map_s16bit_lut_mem_s`: 定义 Map U8C1→S16C1 的查找表内存。
- `xmedia_ive_map_mode_e`: 定义 Map 的模式。
- `xmedia_ive_map_ctrl_s`: 定义 Map 控制参数。
- `xmedia_ive_equalize_hist_ctrl_mem_s`: 定义直方图均衡化辅助内存。
- `xmedia_ive_equalize_hist_ctrl_s`: 定义直方图均衡化控制参数。
- `xmedia_ive_add_ctrl_s`: 定义两图像的加权加控制参数。
- `xmedia_ive_ncc_dst_mem_s`: 定义 NCC 的输出内存信息。
- `xmedia_ive_region_s`: 定义连通区域信息。
- `xmedia_ive_ccblob_s`: 定义连通区域标记的输出信息。
- `xmedia_ive_ccl_mode_e`: 定义连通区域模式。
- `xmedia_ive_ccl_ctrl_s`: 定义连通区域标记控制参数。
- `xmedia_ive_gmm_ctrl_s`: 定义 GMM 背景建模的控制参数。
- `xmedia_ive_gmm2_sns_factor_mode_e`: 定义灵敏度系数模式。
- `xmedia_ive_gmm2_life_update_factor_mode_e`: 定义模型时长参数更新模式。
- `xmedia_ive_gmm2_ctrl_s`: 定义 GMM2 背景建模的控制参数。
- `xmedia_ive_canny_stack_size_s`: 定义 Canny 边缘前半部分计算时强边缘点栈大小结构体。
- `xmedia_ive_canny_hys_edge_ctrl_s`: 定义 Canny 边缘前半部分计算任务的控制参数。
- `xmedia_ive_lbp_cmp_mode_e`: 定义 LBP 计算的比较模式。
- `xmedia_ive_lbp_ctrl_s`: 定义 LBP 纹理计算控制参数。
- `xmedia_ive_norm_grad_out_ctrl_e`: 定义归一化梯度信息计算任务输出控制枚举类型。
- `xmedia_ive_norm_grad_ctrl_s`: 定义归一化梯度信息计算控制参数。
- `xmedia_ive_lk_optical_flow_pyr_out_mode_e`: 定义金字塔 LK 光流计算输出模式。
- `xmedia_ive_lk_optical_flow_pyr_ctrl_s`: 定义金字塔 LK 光流计算控制参数。

- `xmedia_ive_st_max_eig_s`: 定义 Shi-Tomas-like 角点计算时最大角点响应值结构体。
- `xmedia_ive_st_candi_corner_ctrl_s`: 定义 Shi-Tomas-like 候选角点计算控制参数。
- `xmedia_ive_st_corner_info_s`: 定义 Shi-Tomas-like 角点计算输出的角点信息结构体。
- `xmedia_ive_st_corner_ctrl_s`: 定义 Shi-Tomas-like 角点筛选控制参数。
- `xmedia_ive_sad_mode_e`: 定义 SAD 计算模式。
- `xmedia_ive_sad_out_ctrl_e`: 定义 SAD 输出控制模式。
- `xmedia_ive_sad_ctrl_s`: 定义 SAD 控制参数。
- `xmedia_ive_grad_fg_mode_e`: 定义梯度前景计算模式。
- `xmedia_ive_grad_fg_ctrl_s`: 定义计算梯度前景控制参数。
- `xmedia_ive_candi_bg_pix_s`: 定义候选背景模型数据。
- `xmedia_ive_work_bg_pix_s`: 定义工作背景模型数据。
- `xmedia_ive_bg_life_s`: 定义背景生命力数据。
- `xmedia_ive_bg_model_pix_s`: 定义背景模型数据。
- `xmedia_ive_fg_stat_data_s`: 定义前景状态数据。
- `xmedia_ive_bg_stat_data_s`: 定义背景状态数据。
- `xmedia_ive_match_bg_model_ctrl_s`: 定义背景匹配控制参数。
- `xmedia_ive_update_bg_model_ctrl_s`: 定义背景更新控制参数。
- `xmedia_ive_blob_type_e`: 定义 blob 的数据内存排布。
- `xmedia_ive_blob_s`: 定义多个连续存放的 blob 信息。
- `xmedia_ive_src_blob_s`: 定义源序列。
- `xmedia_ive_dst_blob_s`: 定义输出序列。
- `xmedia_ive_resize_csc_letterbox_op_mode_e`: 定义缩放-色彩空间转换-加边框复合任务模式枚举。
- `xmedia_ive_resize_csc_letterbox_csc_mode_e`: 定义缩放-色彩空间转换-加边框复合任务色彩空间转换模式枚举。
- `xmedia_ive_resize_csc_letterbox_ctrl_s`: 定义缩放-色彩空间转换-加边框复合任务控制参数。
- `xmedia_ive_acc_mode_e`: 定义 acc mask 使用模式。

- `xmedia_ive_acc_ctrl_s`: 定义像素累加控制数据。

## 定点数据类型

### 【说明】

定义定点化的数据类型。

### 【定义】

```
typedef xmedia_u8 xmedia_u0q8;
typedef xmedia_u8 xmedia_u1q7;
typedef xmedia_u8 xmedia_u5q3;
typedef xmedia_u8 xmedia_u3q5;
typedef xmedia_u16 xmedia_u0q16;
typedef xmedia_u16 xmedia_u4q12;
typedef xmedia_u16 xmedia_u6q10;
typedef xmedia_u16 xmedia_u8q8;
typedef xmedia_u16 xmedia_u9q7;
typedef xmedia_u16 xmedia_u12q4;
typedef xmedia_u16 xmedia_u14q2;
typedef xmedia_u16 xmedia_u5q11;
typedef xmedia_u16 xmedia_u1q15;
typedef xmedia_u16 xmedia_u2q14;
typedef xmedia_u16 xmedia_ufp16;
/* 8bits unsigned integer,4bits decimal fraction,4bits flag_bits */
typedef xmedia_u16 xmedia_u8q4f4;
typedef xmedia_s16 xmedia_s9q7;
typedef xmedia_s16 xmedia_s14q2;
typedef xmedia_s16 xmedia_s1q15;
typedef xmedia_u32 xmedia_u22q10;
typedef xmedia_u32 xmedia_u25q7;
typedef xmedia_u32 xmedia_u21q11;
typedef xmedia_u32 xmedia_u14q18;
typedef xmedia_u32 xmedia_u8q24;
typedef xmedia_u32 xmedia_u4q28;
typedef xmedia_s32 xmedia_s25q7;
typedef xmedia_s32 xmedia_s16q16;
typedef xmedia_s32 xmedia_s14q18;
typedef xmedia_s32 xmedia_s20q12;
typedef xmedia_s32 xmedia_s24q8;
```

### 【成员】

| 成员名称         | 描述                                                  |
|--------------|-----------------------------------------------------|
| xmedia_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。           |
| xmedia_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。   |
| xmedia_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。   |
| xmedia_u3q5  | 用高 3bit 无符号数据表示整数部分，低 5bit 表示小数部分。文档中用 uq3.5 来表示。   |
| xmedia_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。         |
| xmedia_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。 |
| xmedia_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。 |
| xmedia_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。   |
| xmedia_u9q7  | 用高 9bit 无符号数据表示整数部分，低 7bit 表示小数部分。文档中用 uq9.7 来表示。   |
| xmedia_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。 |
| xmedia_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。 |
| xmedia_u5q11 | 用高 5bit 无符号数据表示整数部分，低 11bit 表示小数部分。文档中用 uq5.11 来表示。 |
| xmedia_u1q15 | 用高 1bit 无符号数据表示整数部分，低 15bit 表示小数部分。文档中用 uq1.15 来表示。 |
| xmedia_u2q14 | 用高 2bit 无符号数据表示整数部分，低 14bit 表示小数部分。文档中用 uq2.14 来表示。 |

| 成员名称          | 描述                                                    |
|---------------|-------------------------------------------------------|
| xmedia_ufp16  | 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。   |
| xmedia_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。     |
| xmedia_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。   |
| xmedia_s1q15  | 用高 1bit 有符号数据表示整数部分，低 15bit 表示小数部分。文档中用 sq1.15 来表示。   |
| xmedia_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。 |
| xmedia_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。   |
| xmedia_u21q11 | 用高 21bit 无符号数据表示整数部分，低 11bit 表示小数部分。文档中用 uq21.11 来表示。 |
| xmedia_u14q18 | 用高 14bit 无符号数据表示整数部分，低 18bit 表示小数部分。文档中用 uq14.18 来表示。 |
| xmedia_u8q24  | 用高 8bit 无符号数据表示整数部分，低 24bit 表示小数部分。文档中用 uq8.24 来表示。   |
| xmedia_u4q28  | 用高 4bit 无符号数据表示整数部分，低 28bit 表示小数部分。文档中用 uq4.28 来表示。   |
| xmedia_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。   |
| xmedia_s16q16 | 用高 16bit 有符号数据表示整数部分，低 16bit 表示小数部分。文档中用 sq16.16 来表示。 |
| xmedia_s14q18 | 用高 14bit 有符号数据表示整数部分，低 18bit 表示小数部分。文档中用 sq14.18 来表示。 |
| xmedia_s20q12 | 用高 20bit 有符号数据表示整数部分，低 12bit 表示小数部分。文档中用 sq20.12 来表示。 |

| 成员名称          | 描述                                                                 |
|---------------|--------------------------------------------------------------------|
| xmedia_s24q8  | 用高 24bit 有符号数据表示整数部分，低 8bit 表示小数部分。文档中用 sq24.8 来表示。                |
| xmedia_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】

xmedia\_uxqyfzlxmedia\_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。

### XMEDIA\_IVE\_HIST\_NUM

#### 【说明】

定义直方图统计 bin 数目。

#### 【定义】

```
#define XMEDIA_IVE_HIST_NUM 256
```

#### 【成员】

无。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

## XMEDIA\_IVE\_MAP\_NUM

### 【说明】

定义映射查找表项数目。

### 【定义】

```
#define XMEDIA_IVE_MAP_NUM 256
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## XMEDIA\_IVE\_MAX\_REGION\_NUM

### 【说明】

定义最大连通区域数目。

### 【定义】

```
#define XMEDIA_IVE_MAX_REGION_NUM 254
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## XMEDIA\_IVE\_ST\_MAX\_CORNER\_NUM

### 【说明】

定义 Shi-Tomasi-like 角点最大数目。

### 【定义】

```
#define XMEDIA_IVE_ST_MAX_CORNER_NUM 500
```

#### 【成员】

无。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_image\_type\_e

#### 【说明】

定义二维广义图像支持的图像类型。

#### 【定义】

```
typedef enum xmedia_ive_image_type_e
{
 XMEDIA_IVE_IMAGE_TYPE_U8C1 = 0x0,
 XMEDIA_IVE_IMAGE_TYPE_S8C1 = 0x1,

 XMEDIA_IVE_IMAGE_TYPE_YUV420SP = 0x2, /*YUV420 SemiPlanar*/
 XMEDIA_IVE_IMAGE_TYPE_YUV422SP = 0x3, /*YUV422 SemiPlanar*/
 XMEDIA_IVE_IMAGE_TYPE_YUV420P = 0x4, /*YUV420 Planar */
 XMEDIA_IVE_IMAGE_TYPE_YUV422P = 0x5, /*YUV422 planar */

 XMEDIA_IVE_IMAGE_TYPE_S8C2_PACKAGE = 0x6,
 XMEDIA_IVE_IMAGE_TYPE_S8C2_PLANAR = 0x7,

 XMEDIA_IVE_IMAGE_TYPE_S16C1 = 0x8,
 XMEDIA_IVE_IMAGE_TYPE_U16C1 = 0x9,

 XMEDIA_IVE_IMAGE_TYPE_U8C3_PACKAGE = 0xa,
 XMEDIA_IVE_IMAGE_TYPE_U8C3_PLANAR = 0xb,

 XMEDIA_IVE_IMAGE_TYPE_S32C1 = 0xc,
 XMEDIA_IVE_IMAGE_TYPE_U32C1 = 0xd,

 XMEDIA_IVE_IMAGE_TYPE_S64C1 = 0xe,
 XMEDIA_IVE_IMAGE_TYPE_U64C1 = 0xf,
```

XMEDIA\_IVE\_IMAGE\_TYPE\_BUTT

}xmedia\_ive\_image\_type\_e;

#### 【成员】

| 成员名称                               | 描述                                                          |
|------------------------------------|-------------------------------------------------------------|
| XMEDIA_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。                      |
| XMEDIA_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。                      |
| XMEDIA_IVE_IMAGE_TYPE_YUV420SP     | YUV420 Semiplanar 格式的图像。<br>请参见图 1-3。                       |
| XMEDIA_IVE_IMAGE_TYPE_YUV422SP     | YUV422 Semiplanar 格式的图像。<br>请参见图 1-4。                       |
| XMEDIA_IVE_IMAGE_TYPE_YUV420P      | YUV420 Planar 格式的图像。请参见图 1-5。                               |
| XMEDIA_IVE_IMAGE_TYPE_YUV422P      | YUV422 Planar 格式的图像。请参见图 1-6。                               |
| XMEDIA_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。<br>请参见图 1-7。 |
| XMEDIA_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。<br>请参见图 1-8。  |
| XMEDIA_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。                      |
| XMEDIA_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。                      |
| XMEDIA_IVE_IMAGE_TYPE_U8C3_PACKAGE | 每个像素用 3 个 8bit 无符号数据表示且以 Package 格式存储 3 通道图像。<br>请参见图 1-9。  |

| 成员名称                              | 描述                                                                 |
|-----------------------------------|--------------------------------------------------------------------|
| XMEDIA_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。<br>请参见图 1-10。 |
| XMEDIA_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。                             |
| XMEDIA_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。                             |
| XMEDIA_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。                             |
| XMEDIA_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】

[xmedia\\_ive\\_image\\_s](#)

[xmedia\\_ive\\_src\\_image\\_s](#)

[xmedia\\_ive\\_dst\\_image\\_s](#)

[xmedia\\_en\\_ive\\_err\\_code\\_e](#)

#### 【说明】

定义错误码。

#### 【定义】

```
typedef enum xmedia_en_ive_err_code_e
{
 ERR_XMEDIA_IVE_SYS_TIMEOUT = 0x40, /* IVE process timeout */
 ERR_XMEDIA_IVE_QUERY_TIMEOUT = 0x41, /* IVE query timeout */
 ERR_XMEDIA_IVE_OPEN_FILE = 0x42, /* IVE open file error */
 ERR_XMEDIA_IVE_READ_FILE = 0x43, /* IVE read file error */
 ERR_XMEDIA_IVE_WRITE_FILE = 0x44, /* IVE write file error */
 ERR_XMEDIA_IVE_BUS_ERR = 0x45,
```

```
ERR_XMEDIA_IVE_BUTT
}xmedia_en_ive_err_code_e;
```

#### 【成员】

| 成员名称                             | 描述          |
|----------------------------------|-------------|
| ERR_XMEDIA_IVE_SYS_TIMEO<br>UT   | 系统超时。       |
| ERR_XMEDIA_IVE_QUERY_TIM<br>EOUT | Query 查询超时。 |
| ERR_XMEDIA_IVE_OPEN_FILE         | 打开文件失败。     |
| ERR_XMEDIA_IVE_READ_FILE         | 读文件失败。      |
| ERR_XMEDIA_IVE_WRITE_FILE        | 写文件失败。      |
| ERR_XMEDIA_IVE_BUS_ERR           | 总线错误。       |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_image\_s

#### 【说明】

定义二维广义图像信息。

#### 【定义】

```
typedef struct xmedia_ive_image_s
{
 xmedia_u64 au64phyaddr[3]; /* RW;The physical address of the image */
 xmedia_u64 au64viraddr[3]; /* RW;The virtual address of the image */
 xmedia_u32 au32stride[3]; /* RW;The stride of the image */
 xmedia_u32 u32width; /* RW;The width of the image */
 xmedia_u32 u32height; /* RW;The height of the image */
 xmedia_ive_image_type_e en_type; /* RW;The type of the image */
}xmedia_ive_image_s;
```

#### 【成员】

| 成员名称           | 描述           |
|----------------|--------------|
| au64phyaddr[3] | 广义图像的物理地址数组。 |
| au64viraddr[3] | 广义图像的虚拟地址数组。 |
| au32stride[3]  | 广义图像的跨度。     |
| u32width       | 广义图像的宽度。     |
| u32height      | 广义图像的高度。     |
| en_type        | 广义图像的图像类型。   |

#### 【注意事项】

- 不同的算子对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】

[xmedia\\_ive\\_image\\_type\\_e](#)

[xmedia\\_ive\\_src\\_image\\_s](#)

[xmedia\\_ive\\_dst\\_image\\_s](#)

### xmedia\_ive\_src\_image\_s

#### 【说明】

定义源图像。

#### 【定义】

```
typedef xmedia_ive_image_s xmedia_ive_src_image_s;
```

#### 【成员】

无。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_image\\_s](#)

[xmedia\\_ive\\_dst\\_image\\_s](#)

## xmedia\_ive\_dst\_image\_s

### 【说明】

定义输出图像。

### 【定义】

```
typedef xmedia_ive_image_s xmedia_ive_dst_image_s;
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

[xmedia\\_ive\\_image\\_s](#)

[xmedia\\_ive\\_src\\_image\\_s](#)

## xmedia\_ive\_data\_s

### 【说明】

定义以 byte 为单位的二维数据信息。

### 【定义】

对于 XM72050200:

```
typedef struct xmedia_ive_data_s
{
 xmedia_u64 u64PhyAddr; /* RW;The physical address of the data */
 xmedia_u64 u64VirAddr; /* RW;The virtaul address of the data */

 xmedia_u32 u32Stride; /* RW;The stride of 2D data by byte */
 xmedia_u32 u32Width; /* RW;The width of 2D data by byte */
 xmedia_u32 u32Height; /* RW;The height of 2D data by byte */

 xmedia_u32 u32Reserved;
}xmedia_ive_data_s;
```

### 【成员】

对于 XM72050200:

| 成员名称        | 描述      |
|-------------|---------|
| u64PhyAddr  | 图像物理地址。 |
| u64VirAddr  | 图像虚拟地址。 |
| u32Stride   | 图像跨度。   |
| u32Height   | 图像高度。   |
| u32Width    | 图像宽度。   |
| u32Reserved | 保留位。    |

**【注意事项】**

表示以 byte 为单位的二维数据；可以与 [xmedia\\_ive\\_image\\_s](#) 图像进行转换。

**【相关数据类型及接口】**

无。

## xmedia\_ive\_src\_data\_s

**【说明】**

定义以 byte 为单位的二维源数据信息。

**【定义】**

```
typedef xmedia_ive_data_s xmedia_ive_src_data_s;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

[xmedia\\_ive\\_image\\_s](#)

[xmedia\\_ive\\_dst\\_data\\_s](#)

## xmedia\_ive\_dst\_data\_s

### 【说明】

定义 byte 为单位的二维输出数据信息。

### 【定义】

```
typedef xmedia_ive_data_s xmedia_ive_dst_data_s;
```

### 【成员】

无。

### 【注意事项】

无。

### 【相关数据类型及接口】

[xmedia\\_ive\\_image\\_s](#)

[xmedia\\_ive\\_src\\_image\\_s](#)

## xmedia\_ive\_mem\_info\_s

### 【说明】

定义一维数据内存信息。

### 【定义】

```
typedef struct xmedia_ive_mem_info_s
{
 xmedia_u64 u64phyaddr; /* RW;The physical address of the memory */
 xmedia_u64 u64viraddr; /* RW;The virtual address of the memory */
 xmedia_u32 u32size; /* RW;The size of memory */
}xmedia_ive_mem_info_s;
```

### 【成员】

| 成员名称       | 描述            |
|------------|---------------|
| u64phyaddr | 一维数据物理地址。     |
| u64viraddr | 一维数据虚拟地址。     |
| u32size    | 一维数据 byte 数目。 |

**【注意事项】**

无。

**【相关数据类型及接口】**

[xmedia\\_ive\\_src\\_mem\\_info\\_s](#)

[xmedia\\_ive\\_dst\\_mem\\_info\\_s](#)

## xmedia\_ive\_src\_mem\_info\_s

**【说明】**

定义一维源数据。

**【定义】**

```
typedef xmedia_ive_mem_info_s xmedia_ive_src_mem_info_s;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

[xmedia\\_ive\\_mem\\_info\\_s](#)

[xmedia\\_ive\\_dst\\_mem\\_info\\_s](#)

## xmedia\_ive\_dst\_mem\_info\_s

**【说明】**

定义一维输出数据。

**【定义】**

```
typedef xmedia_ive_mem_info_s xmedia_ive_dst_mem_info_s;
```

**【成员】**

无。

**【注意事项】**

无。

**【相关数据类型及接口】**

[xmedia\\_ive\\_mem\\_info\\_s](#)  
[xmedia\\_ive\\_src\\_mem\\_info\\_s](#)

## xmedia\_ive\_8bit\_u

### 【说明】

定义 8bit 数据联合体。

### 【定义】

```
typedef union xmedia_ive_8bit_u
{
 xmedia_s8 s8val;
 xmedia_u8 u8val;
}xmedia_ive_8bit_u;
```

### 【成员】

| 成员名称  | 描述          |
|-------|-------------|
| s8val | 有符号 8bit 值。 |
| u8val | 无符号 8bit 值。 |

### 【注意事项】

无

### 【相关数据类型及接口】

无。

## xmedia\_ive\_point\_u16\_s

### 【说明】

定义 U16 表示的点信息结构体。

### 【定义】

```
typedef struct xmedia_ive_point_u16_s
{
 xmedia_u16 u16x;
 xmedia_u16 u16y;
}xmedia_ive_point_u16_s;
```

#### 【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

xmedia\_ive\_point\_s16\_s

#### 【说明】

定义 S16 表示的点信息结构体。

#### 【定义】

```
typedef struct xmedia_ive_point_s16_s
{
 xmedia_s16 s16x;
 xmedia_s16 s16y;
}xmedia_ive_point_s16_s;
```

#### 【成员】

| 成员名称 | 描述       |
|------|----------|
| s16x | 点的 x 坐标。 |
| s16y | 点的 y 坐标。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_point\_u14q2\_s

**【说明】**

定义 U14Q2 坐标结构体。

**【定义】**

```
typedef struct xmedia_ive_point_u14q2_s
{
 xmedia_u14q2 u14q2X;
 xmedia_u14q2 u14q2Y;
}xmedia_ive_point_u14q2_s;
```

**【成员】**

| 成员名称   | 描述       |
|--------|----------|
| u14q2x | 点的 x 坐标。 |
| u14q2y | 点的 y 坐标。 |

**【注意事项】**

无。

**【相关数据类型及接口】**

无。

### xmedia\_ive\_point\_s25q7\_s

**【说明】**

定义 S25Q7 定点表示的点信息结构体。

**【定义】**

```
typedef struct xmedia_ive_point_s25q7_s
{
 xmedia_s25q7 s25q7x; /*X coordinate*/
 xmedia_s25q7 s25q7y; /*Y coordinate*/
}xmedia_ive_point_s25q7_s;
```

**【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。

### xmedia\_ive\_rect\_u16\_s

#### 【说明】

定义 U16 表示的矩形信息结构体。

#### 【定义】

```
typedef struct xmedia_ive_rect_u16_s
{
 xmedia_u16 u16x;
 xmedia_u16 u16y;
 xmedia_u16 u16width;
 xmedia_u16 u16height;
} xmedia_ive_rect_u16_s;
```

#### 【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。

### xmedia\_ive\_rect\_u32\_s

#### 【说明】

定义 U32 表示的矩形信息结构体。

#### 【定义】

```
typedef struct xmedia_ive_rect_u32_s
{
 xmedia_u32 u32x;
 xmedia_u32 u32y;
 xmedia_u32 u32width;
 xmedia_u32 u32height;
}xmedia_ive_rect_u32_s;
```

#### 【成员】

| 成员名称      | 描述                  |
|-----------|---------------------|
| u32x      | 矩形相对于坐标原点最近点的 x 坐标。 |
| u32y      | 矩形相对于坐标原点最近点的 y 坐标。 |
| u32width  | 矩形的宽。               |
| u32height | 矩形的高。               |

#### 【注意事项】

无

#### 【相关数据类型及接口】

无

### xmedia\_ive\_rect\_s24q8\_s

#### 【说明】

定义 S24Q8 表示的矩形信息结构体。

#### 【定义】

```
typedef struct xmedia_ive_rect_s24q8_s
{
 xmedia_s24q8 s24q8x;
 xmedia_s24q8 s24q8y;
 xmedia_u32 u32width;
 xmedia_u32 u32height;
}xmedia_ive_rect_s24q8_s;
```

#### 【成员】

| 成员名称      | 描述                  |
|-----------|---------------------|
| s24q8x    | 矩形相对于坐标原点最近点的 x 坐标。 |
| s24q8y    | 矩形相对于坐标原点最近点的 y 坐标。 |
| u32width  | 矩形的宽。               |
| u32height | 矩形的高。               |

#### 【注意事项】

无

#### 【相关数据类型及接口】

无

xmedia\_ive\_dma\_mode\_e

#### 【说明】

定义 DMA 操作模式。

#### 【定义】

```
typedef enum xmedia_ive_dma_mode_e
{
 XMEDIA_IVE_DMA_MODE_DIRECT_COPY = 0x0,
 XMEDIA_IVE_DMA_MODE_INTERVAL_COPY = 0x1,
 XMEDIA_IVE_DMA_MODE_SET_3BYTE = 0x2,
 XMEDIA_IVE_DMA_MODE_SET_8BYTE = 0x3,
 XMEDIA_IVE_DMA_MODE_BUTT
}xmedia_ive_dma_mode_e;
```

#### 【成员】

| 成员名称                              | 描述                                                    |
|-----------------------------------|-------------------------------------------------------|
| XMEDIA_IVE_DMA_MODE_DIRECT_COPY   | 直接快速拷贝模式。                                             |
| XMEDIA_IVE_DMA_MODE_INTERVAL_COPY | 间隔拷贝模式，请参见 <a href="#">xmedia_ive_dma</a> 【注意】说明。     |
| XMEDIA_IVE_DMA_MODE_SET_3BYTE     | 3byte 赋值模式，请参见 <a href="#">xmedia_ive_dma</a> 【注意】说明。 |
| XMEDIA_IVE_DMA_MODE_SET_8BYTE     | 8byte 赋值模式，请参见 <a href="#">xmedia_ive_dma</a> 【注意】说明。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_dma\\_ctrl\\_s](#)

### xmedia\_ive\_dma\_ctrl\_s

#### 【说明】

定义 DMA 控制信息。

#### 【定义】

```
typedef struct xmedia_ive_dma_ctrl_s
{
 xmedia_ive_dma_mode_e en_mode;
 xmedia_u64 u64_val;
 xmedia_u8 u8_hor_seg_size;
 xmedia_u8 u8_elem_size;
 xmedia_u16 u16_ver_seg_rows;
}xmedia_ive_dma_ctrl_s;
```

#### 【成员】

| 成员名称             | 描述                                                                                                                                     |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| en_mode          | DMA 操作模式。                                                                                                                              |
| u64_val          | 仅赋值模式使用，用于对内存赋值，3byte 赋值模式用低 3byte 保存。                                                                                                 |
| u8_hor_seg_size  | 仅间隔拷贝模式使用，水平方向将源图像一行分割的段大小。<br>取值范围：{2, 3, 4, 8, 16}。                                                                                  |
| u8_elem_size     | 仅间隔拷贝模式使用，分割的每一段中前 u8_elem_sizebyte 为有效的拷贝字段。<br>取值范围：[1, u8_hor_seg_size-1]。                                                          |
| u16_ver_seg_rows | 仅间隔拷贝模式使用，将每 u16_ver_seg_rows 行中第一行数据分割为 u8_hor_seg_size 大小的段，拷贝每段中的前 u8_elem_size 大小的字节<br>取值范围：[1, min{65535/srcStride, srcHeight}]。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_dma\\_mode\\_e](#)

[xmedia\\_ive\\_filter\\_ctrl\\_s](#)

#### 【说明】

定义模板滤波控制信息。

#### 【定义】

```
typedef struct xmedia_ive_filter_ctrl_s
{
 xmedia_s8 as8_mask[25]; /*Template parameter filter coefficient*/
 xmedia_u8 u8_norm; /*Normalization parameter, by right shift*/
}xmedia_ive_filter_ctrl_s;
```

#### 【成员】

| 成员名称        | 描述                              |
|-------------|---------------------------------|
| as8_mask[5] | 5x5 模板系数，外围系数设为 0 可实现 3x3 模板滤波。 |
| u8_norm     | 归一化参数。<br>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。

### xmedia\_ive\_csc\_mode\_e

#### 【说明】

定义色彩空间转换模式。

#### 【定义】

```
typedef enum xmedia_ive_csc_mode_e
{
/*CSC: YUV2RGB, video transfer mode, RGB value range [16, 235]*/
XMEDIA_IVE_CSC_MODE_VIDEO_BT601_YUV2RGB = 0x0,
/*CSC: YUV2RGB, video transfer mode, RGB value range [16, 235]*/
XMEDIA_IVE_CSC_MODE_VIDEO_BT709_YUV2RGB = 0x1,
/*CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2RGB = 0x2,
/*CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2RGB = 0x3,
/*CSC: YUV2HSV, picture transfer mode, HSV value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2HSV = 0x4,
/*CSC: YUV2HSV, picture transfer mode, HSV value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2HSV = 0x5,
/*CSC: YUV2LAB, picture transfer mode, Lab value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2LAB = 0x6,
/*CSC: YUV2LAB, picture transfer mode, Lab value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2LAB = 0x7,
/*CSC: RGB2YUV, video transfer mode, YUV value range [0, 255]*/
```

```

XMEDIA_IVE_CSC_MODE_VIDEO_BT601_RGB2YUV = 0x8,
/*CSC: RGB2YUV, video transfer mode, YUV value range [0, 255]*/
XMEDIA_IVE_CSC_MODE_VIDEO_BT709_RGB2YUV = 0x9,
/*CSC: RGB2YUV, picture transfer mode, Y:[16, 235],UV:[16, 240]*/
XMEDIA_IVE_CSC_MODE_PIC_BT601_RGB2YUV = 0xa,
/*CSC: RGB2YUV, picture transfer mode, Y:[16, 235],UV:[16, 240]*/
XMEDIA_IVE_CSC_MODE_PIC_BT709_RGB2YUV = 0xb,

XMEDIA_IVE_CSC_MODE_BUTT
}xmedia_ive_csc_mode_e;

```

#### 【成员】

| 成员名称                                    | 描述                    |
|-----------------------------------------|-----------------------|
| XMEDIA_IVE_CSC_MODE_VIDEO_BT601_YUV2RGB | BT601 的 YUV2RGB 视频变换。 |
| XMEDIA_IVE_CSC_MODE_VIDEO_BT709_YUV2RGB | BT709 的 YUV2RGB 视频变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2RGB   | BT601 的 YUV2RGB 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2RGB   | BT709 的 YUV2RGB 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2HSV   | BT601 的 YUV2HSV 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2HSV   | BT709 的 YUV2HSV 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT601_YUV2LAB   | BT601 的 YUV2LAB 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT709_YUV2LAB   | BT709 的 YUV2LAB 图像变换。 |
| XMEDIA_IVE_CSC_MODE_VIDEO_BT601_RGB2YUV | BT601 的 RGB2YUV 视频变换。 |
| XMEDIA_IVE_CSC_MODE_VIDEO_BT709_RGB2YUV | BT709 的 RGB2YUV 视频变换。 |

| 成员名称                                  | 描述                    |
|---------------------------------------|-----------------------|
| XMEDIA_IVE_CSC_MODE_PIC_BT601_RGB2YUV | BT601 的 RGB2YUV 图像变换。 |
| XMEDIA_IVE_CSC_MODE_PIC_BT709_RGB2YUV | BT709 的 RGB2YUV 图像变换。 |

#### 【注意事项】

- XMEDIA\_IVE\_CSC\_MODE\_VIDEO\_BT601\_YUV2RGB 和 XMEDIA\_IVE\_CSC\_MODE\_VIDEO\_BT709\_YUV2RGB 模式，输出满足  $16 \leq R, G, B \leq 235$ 。
- XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT601\_YUV2RGB 和 XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT709\_YUV2RGB 模式，输出满足  $0 \leq R, G, B \leq 255$ 。
- XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT601\_YUV2HSV 和 XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT709\_YUV2HSV 模式，输出满足  $0 \leq H, S, V \leq 255$ 。
- XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT601\_YUV2LAB 和 XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT709\_YUV2LAB 模式，输出满足  $0 \leq L, A, B \leq 255$ 。
- XMEDIA\_IVE\_CSC\_MODE\_VIDEO\_BT601\_RGB2YUV 和 XMEDIA\_IVE\_CSC\_MODE\_VIDEO\_BT709\_RGB2YUV 模式，输出满足  $0 \leq Y, U, V \leq 255$ 。
- XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT601\_RGB2YUV 和 XMEDIA\_IVE\_CSC\_MODE\_PIC\_BT709\_RGB2YUV 模式，输出满足  $0 \leq Y \leq 235, 0 \leq U, V \leq 240$ 。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_csc\\_ctrl\\_s](#)

[xmedia\\_ive\\_filter\\_and\\_csc\\_ctrl\\_s](#)

xmedia\_ive\_csc\_ctrl\_s

#### 【说明】

定义色彩空间转换控制信息。

### 【定义】

```
typedef struct xmedia_ive_csc_ctrl_s
{
 xmedia_ive_csc_mode_e en_mode; /*Working mode*/
}xmedia_ive_csc_ctrl_s;
```

### 【成员】

| 成员名称    | 描述    |
|---------|-------|
| en_mode | 工作模式。 |

### 【注意事项】

无。

### 【相关数据类型及接口】

xmedia\_ive\_csc\_mode\_e

xmedia\_ive\_filter\_and\_csc\_ctrl\_s

### 【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 xmedia_ive_filter_and_csc_ctrl_s
{
 xmedia_ive_csc_mode_e en_mode; /*CSC working mode*/
 xmedia_s8 as8_mask [25]; /*Template parameter filter coefficient*/
 xmedia_u8 u8_norm; /*Normalization parameter, by right shift*/
}xmedia_ive_filter_and_csc_ctrl_s;
```

### 【成员】

| 成员名称          | 描述                      |
|---------------|-------------------------|
| en_mode       | 工作模式。                   |
| as8_mask [25] | 5x5 模板系数。               |
| u8_norm       | 归一化参数。<br>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】

[xmedia\\_ive\\_csc\\_mode\\_e](#)

xmedia\_ive\_sobel\_out\_ctrl\_e

#### 【说明】

定义 Sobel 输出控制信息。

#### 【定义】

```
typedef enum xmedia_ive_sobel_out_ctrl_e
{
 XMEDIA_IVE_SOBEL_OUT_CTRL_BOTH = 0x0, /*Output horizontal and vertical*/
 XMEDIA_IVE_SOBEL_OUT_CTRL_HOR = 0x1, /*Output horizontal*/
 XMEDIA_IVE_SOBEL_OUT_CTRL_VER = 0x2, /*Output vertical*/
 XMEDIA_IVE_SOBEL_OUT_CTRL_BUTT
}xmedia_ive_sobel_out_ctrl_e;
```

#### 【成员】

| 成员名称                           | 描述                 |
|--------------------------------|--------------------|
| XMEDIA_IVE_SOBEL_OUT_CTRL_BOTH | 同时输出用模板和转置模板滤波的结果。 |
| XMEDIA_IVE_SOBEL_OUT_CTRL_HOR  | 仅输出用模板直接滤波的结果。     |
| XMEDIA_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】

[xmedia\\_ive\\_sobel\\_ctrl\\_s](#)

xmedia\_ive\_sobel\_ctrl\_s

#### 【说明】

定义 Sobel-like 梯度计算控制信息。

#### 【定义】

```
typedef struct xmedia_ive_sobel_ctrl_s
{
 xmedia_ive_sobel_out_ctrl_e en_out_ctrl; /*Output format*/
 xmedia_s8 as8_mask[25]; /*Template parameter*/
}xmedia_ive_sobel_ctrl_s;
```

#### 【成员】

| 成员名称         | 描述        |
|--------------|-----------|
| en_out_ctrl  | 输出控制枚举参数。 |
| as8_mask[25] | 5x5 模板系数。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_sobel\\_out\\_ctrl\\_e](#)

[xmedia\\_ive\\_mag\\_and\\_ang\\_out\\_ctrl\\_e](#)

#### 【说明】

定义梯度幅值与角度计算的输出格式。

#### 【定义】

```
typedef enum xmedia_ive_mag_and_ang_out_ctrl_e
{
 XMEDIA_IVE_MAG_AND_ANG_OUT_CTRL_MAG = 0x0,
 XMEDIA_IVE_MAG_AND_ANG_OUT_CTRL_MAG_AND_ANG = 0x1,
 XMEDIA_IVE_MAG_AND_ANG_OUT_CTRL_BUTT
}xmedia_ive_mag_and_ang_out_ctrl_e;
```

#### 【成员】

| 成员名称                                | 描述     |
|-------------------------------------|--------|
| XMEDIA_IVE_MAG_AND_ANG_OUT_CTRL_MAG | 仅输出幅值。 |

| 成员名称                                        | 描述          |
|---------------------------------------------|-------------|
| XMEDIA_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】

[xmedia\\_ive\\_mag\\_and\\_ang\\_ctrl\\_s](#)

xmedia\_ive\_mag\_and\_ang\_ctrl\_s

#### 【说明】

定义梯度幅值和幅角计算的控制信息。

#### 【定义】

```
typedef struct xmedia_ive_mag_and_ang_ctrl_s
{
 xmedia_ive_mag_and_ang_out_ctrl_e en_out_ctrl;
 xmedia_u16 u16_thr;
 xmedia_s8 as8_mask[25]; /*Template parameter.*/
}xmedia_ive_mag_and_ang_ctrl_s;
```

#### 【成员】

| 成员名称         | 描述             |
|--------------|----------------|
| en_out_ctrl  | 输出格式。          |
| u16_thr      | 用于对幅值进行阈值化的阈值。 |
| as8_mask[25] | 5x5 模板系数。      |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_mag\\_and\\_ang\\_out\\_ctrl\\_e](#)

## xmedia\_ive\_dilate\_ctrl\_s

### 【说明】

定义膨胀控制信息。

### 【定义】

```
typedef struct xmedia_ive_dilate_ctrl_s
{
 xmedia_u8 au8_mask[25]; /*The template parameter value must be 0 or 255.*/
}xmedia_ive_dilate_ctrl_s;
```

### 【成员】

| 成员名称         | 描述                         |
|--------------|----------------------------|
| au8_mask[25] | 5x5 模板系数。<br>取值范围：0 或 255。 |

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## xmedia\_ive\_erode\_ctrl\_s

### 【说明】

定义腐蚀控制信息。

### 【定义】

```
typedef struct xmedia_ive_erode_ctrl_s
{
 xmedia_u8 au8_mask[25]; /*The template parameter value must be 0 or 255.*/
}xmedia_ive_erode_ctrl_s;
```

### 【成员】

| 成员名称         | 描述                         |
|--------------|----------------------------|
| au8_mask[25] | 5x5 模板系数。<br>取值范围：0 或 255。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_thresh\_mode\_e

#### 【说明】

定义图像二值化输出格式。

#### 【定义】

```
typedef enum xmedia_ive_thresh_mode_e
{
 XMEDIA_IVE_THRESH_MODE_BINARY = 0x0, /*srcVal <= lowThr, dstVal =
minVal; srcVal > lowThr, dstVal = maxVal.*/
 XMEDIA_IVE_THRESH_MODE_TRUNC = 0x1, /*srcVal <= lowThr, dstVal =
srcVal; srcVal > lowThr, dstVal = maxVal.*/
 XMEDIA_IVE_THRESH_MODE_TO_MINVAL = 0x2, /*srcVal <= lowThr, dstVal =
minVal; srcVal > lowThr, dstVal = srcVal.*/
 XMEDIA_IVE_THRESH_MODE_MIN_MID_MAX = 0x3, /*srcVal <= lowThr, dstVal =
minVal; lowThr < srcVal <= highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal.*/
 XMEDIA_IVE_THRESH_MODE_ORI_MID_MAX = 0x4, /*srcVal <= lowThr, dstVal =
srcVal; lowThr < srcVal <= highThr, dstVal = midVal; srcVal > highThr, dstVal = maxVal.*/
 XMEDIA_IVE_THRESH_MODE_MIN_MID_ORI = 0x5, /*srcVal <= lowThr, dstVal =
minVal; lowThr < srcVal <= highThr, dstVal = midVal; srcVal > highThr, dstVal = srcVal.*/
 XMEDIA_IVE_THRESH_MODE_MIN_ORI_MAX = 0x6, /*srcVal <= lowThr, dstVal =
minVal; lowThr < srcVal <= highThr, dstVal = srcVal; srcVal > highThr, dstVal = maxVal.*/
 XMEDIA_IVE_THRESH_MODE_ORI_MID_ORI = 0x7, /*srcVal <= lowThr, dstVal =
srcVal; lowThr < srcVal <= highThr, dstVal = midVal; srcVal > highThr, dstVal = srcVal.*/

 XMEDIA_IVE_THRESH_MODE_BUTT
}xmedia_ive_thresh_mode_e;
```

#### 【成员】

| 成员名称                               | 描述                                                                                                                               |
|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_THRESH_MODE_BINARY      | srcVal $\leq$ lowThr, dstVal = minVal; srcVal > lowThr, dstVal = maxVal。                                                         |
| XMEDIA_IVE_THRESH_MODE_TRUNC       | srcVal $\leq$ lowThr, dstVal = srcVal;<br>srcVal > lowThr, dstVal = maxVal。                                                      |
| XMEDIA_IVE_THRESH_MODE_TO_MINVAL   | srcVal $\leq$ lowThr, dstVal = minVal;<br>srcVal > lowThr, dstVal = srcVal。                                                      |
| XMEDIA_IVE_THRESH_MODE_MIN_MID_MAX | srcVal $\leq$ lowThr, dstVal = minVal;<br>lowThr < srcVal $\leq$ highThr, dstVal = midVal;<br>srcVal > highThr, dstVal = maxVal。 |
| XMEDIA_IVE_THRESH_MODE_ORI_MID_MAX | srcVal $\leq$ lowThr, dstVal = srcVal;<br>lowThr < srcVal $\leq$ highThr, dstVal = midVal;<br>srcVal > highThr, dstVal = maxVal。 |
| XMEDIA_IVE_THRESH_MODE_MIN_MID_ORI | srcVal $\leq$ lowThr, dstVal = minVal;<br>lowThr < srcVal $\leq$ highThr, dstVal = midVal;<br>srcVal > highThr, dstVal = srcVal。 |
| XMEDIA_IVE_THRESH_MODE_MIN_ORI_MAX | srcVal $\leq$ lowThr, dstVal = minVal;<br>lowThr < srcVal $\leq$ highThr, dstVal = srcVal;<br>srcVal > highThr, dstVal = maxVal。 |

| 成员名称                               | 描述                                                                                                                     |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_THRESH_MODE_ORI_MID_ORI | srcVal ≤ lowThr, dstVal = srcVal;<br>lowThr < srcVal ≤ highThr, dstVal = midVal;<br>srcVal > highThr, dstVal = srcVal。 |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_thresh](#) 中的【注意】，示意图请参见图 2-8。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_ctrl\\_s](#)

xmedia\_ive\_thresh\_ctrl\_s

#### 【说明】

定义图像二值化控制信息。

#### 【定义】

```
typedef struct xmedia_ive_thresh_ctrl_s
{
 xmedia_ive_thresh_mode_e en_mode;
 xmedia_u8 u8_low_thr; /*user-defined threshold, 0<=u8_low_thr<=255 */
 xmedia_u8 u8_high_thr; /*user-defined threshold, if
en_mode<XMEDIA_IVE_THRESH_MODE_MIN_MID_MAX, u8_high_thr is not used, else
0<=u8_low_thr<=u8_high_thr<=255;*/
 xmedia_u8 u8_min_val; /*Minimum value when tri-level thresholding*/
 xmedia_u8 u8_mid_val; /*Middle value when tri-level thresholding, if en_mode<2,
u32MidVal is not used; */
 xmedia_u8 u8_max_val; /*Maxmum value when tri-level thresholding*/
}xmedia_ive_thresh_ctrl_s;
```

#### 【成员】

| 成员名称    | 描述       |
|---------|----------|
| en_mode | 阈值化运算模式。 |

| 成员名称        | 描述                                                                |
|-------------|-------------------------------------------------------------------|
| u8_low_thr  | 低阈值。<br>取值范围：[0,255]。                                             |
| u8_high_thr | 高阈值。<br>$0 \leq u8\_low\_thresh \leq u8\_high\_thresh \leq 255$ 。 |
| u8_min_val  | 最小值。<br>取值范围：[0,255]。                                             |
| u8_mid_val  | 中间值。<br>取值范围：[0,255]。                                             |
| u8_max_val  | 最大值。<br>取值范围：[0,255]。                                             |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_mode\\_e](#)

[xmedia\\_ive\\_sub\\_mode\\_e](#)

#### 【说明】

定义两图像相减输出格式。

#### 【定义】

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

#### 【成员】

| 成员名称                      | 描述               |
|---------------------------|------------------|
| XMEDIA_IVE_SUB_MODE_ABS   | 取差的绝对值。          |
| XMEDIA_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】

[xmedia\\_ive\\_sub\\_ctrl\\_s](#)

### xmedia\_ive\_sub\_ctrl\_s

#### 【说明】

定义两图像相减控制参数。

#### 【定义】

```
typedef struct xmedia_ive_sub_ctrl_s
{
 xmedia_ive_sub_mode_e en_mode;
}xmedia_ive_sub_ctrl_s;
```

#### 【成员】

| 成员名称    | 描述       |
|---------|----------|
| en_mode | 两图像相减模式。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_sub\\_mode\\_e](#)

### xmedia\_ive\_integ\_out\_ctrl\_e

#### 【说明】

定义积分图输出控制参数。

### 【定义】

```
typedef enum xmedia_ive_integ_out_ctrl_e
{
 XMEDIA_IVE_INTEG_OUT_CTRL_COMBINE = 0x0,
 XMEDIA_IVE_INTEG_OUT_CTRL_SUM = 0x1,
 XMEDIA_IVE_INTEG_OUT_CTRL_SQSUM = 0x2,
 XMEDIA_IVE_INTEG_OUT_CTRL_BUTT
}xmedia_ive_integ_out_ctrl_e;
```

### 【成员】

| 成员名称                              | 描述                   |
|-----------------------------------|----------------------|
| XMEDIA_IVE_INTEG_OUT_CTRL_COMBINE | 和、平方和积分图组合输出，如图1-13。 |
| XMEDIA_IVE_INTEG_OUT_CTRL_SUM     | 仅和积分图输出。             |
| XMEDIA_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】

[xmedia\\_ive\\_integ\\_ctrl\\_s](#)

xmedia\_ive\_integ\_ctrl\_s

### 【说明】

定义积分图计算控制参数。

### 【定义】

```
typedef struct xmedia_ive_integ_ctrl_s
{
 xmedia_ive_integ_out_ctrl_e en_out_ctrl;
}xmedia_ive_integ_ctrl_s;
```

### 【成员】

| 成员名称        | 描述         |
|-------------|------------|
| en_out_ctrl | 积分图输出控制参数。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_integ\\_out\\_ctrl\\_e](#)

[xmedia\\_ive\\_thresh\\_s16\\_mode\\_e](#)

#### 【说明】

定义 16bit 有符号图像的阈值化模式。

#### 【定义】

```
typedef enum xmedia_ive_thresh_s16_mode_e
{
 XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX = 0x0,
 XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX = 0x1,
 XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX = 0x2,
 XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX = 0x3,
 XMEDIA_IVE_THRESH_S16_MODE_BUTT
}xmedia_ive_thresh_s16_mode_e;
```

#### 【成员】

| 成员名称                                             | 描述                                                                                                                                                                                                                                                 |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_MID_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{midVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |

| 成员名称                                             | 描述                                                                                                                                                                                                                                                 |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_THRESH_S16_MODE_S16_TO_S8_MIN_ORI_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{srcVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |
| XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_MID_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{midVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |
| XMEDIA_IVE_THRESH_S16_MODE_S16_TO_U8_MIN_ORI_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{srcVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_thresh\\_s16](#) 中的【注意】，示意图请参见图 2-9。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_s16\\_ctrl\\_s](#)

[xmedia\\_ive\\_thresh\\_s16\\_ctrl\\_s](#)

#### 【说明】

定义 16bit 有符号图像的阈值化控制参数。

#### 【定义】

```
typedef struct xmedia_ive_thresh_s16_ctrl_s
{
 xmedia_ive_thresh_s16_mode_e en_mode;
 xmedia_s16 s16_low_thr; /*user-defined threshold*/
}
```

```

xmedia_s16 s16_high_thr; /*user-defined threshold*/
xmedia_ive_8bit_u un8_min_val; /*Minimum value when tri-level thresholding*/
xmedia_ive_8bit_u un8_mid_val; /*Middle value when tri-level thresholding*/
xmedia_ive_8bit_u un8_max_val; /*Maxmum value when tri-level thresholding*/
}xmedia_ive_thresh_s16_ctrl_s;

```

#### 【成员】

| 成员名称         | 描述       |
|--------------|----------|
| en_mode      | 阈值化运算模式。 |
| s16_low_thr  | 低阈值。     |
| s16_high_thr | 高阈值。     |
| un8_min_val  | 最小值。     |
| un8_mid_val  | 中间值。     |
| un8_max_val  | 最大值。     |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_thresh\\_s16](#) 中的【注意】，示意图请参见图 2-9。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_s16\\_mode\\_e](#)

[xmedia\\_ive\\_thresh\\_u16\\_mode\\_e](#)

#### 【说明】

定义 16bti 无符号图像的阈值化模式。

#### 【定义】

```

typedef enum xmedia_ive_thresh_u16_mode_e
{
 XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX = 0x0,
 XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX = 0x1,
 XMEDIA_IVE_THRESH_U16_MODE_BUTT
}xmedia_ive_thresh_u16_mode_e;

```

#### 【成员】

| 成员名称                                             | 描述                                                                                                                                                                                                                                                 |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_MID_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{midVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |
| XMEDIA_IVE_THRESH_U16_MODE_U16_TO_U8_MIN_ORI_MAX | $\text{srcVal} \leq \text{lowThr}$ ,<br>$\text{dstVal} = \text{minVal}$ ;<br>$\text{lowThr} < \text{srcVal} \leq \text{highThr}$ ,<br>$\text{dstVal} = \text{srcVal}$ ;<br>$\text{srcVal} > \text{highThr}$ ,<br>$\text{dstVal} = \text{maxVal}$ ; |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_thresh\\_u16](#) 中的【注意】，示意图请参见图 2-10。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_u16\\_ctrl\\_s](#)

[xmedia\\_ive\\_thresh\\_u16\\_ctrl\\_s](#)

#### 【说明】

定义 16bit 无符号图像的阈值化控制参数。

#### 【定义】

```
typedef struct xmedia_ive_thresh_u16_ctrl_s
{
 xmedia_ive_thresh_u16_mode_e en_mode;
 xmedia_u16 u16_low_thr;
 xmedia_u16 u16_high_thr;
 xmedia_u8 u8_min_val;
 xmedia_u8 u8_mid_val;
 xmedia_u8 u8_max_val;
}xmedia_ive_thresh_u16_ctrl_s;
```

#### 【成员】

| 成员名称         | 描述                    |
|--------------|-----------------------|
| en_mode      | 阈值化运算模式。              |
| u16_low_thr  | 低阈值。                  |
| u16_high_thr | 高阈值。                  |
| u8_min_val   | 最小值。<br>取值范围：[0,255]。 |
| u8_mid_val   | 中间值。<br>取值范围：[0,255]。 |
| u8_max_val   | 最大值。<br>取值范围：[0,255]。 |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_thresh\\_u16](#) 中的【注意】，示意图请参见图 2-10。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_thresh\\_u16\\_mode\\_e](#)

[xmedia\\_ive\\_16bit\\_to\\_8bit\\_mode\\_e](#)

#### 【说明】

定义 16bit 图像数据到 8bit 图像数据的转化模式。

#### 【定义】

```
typedef enum xmedia_ive_16bit_to_8bit_mode_e
{
 XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8 = 0x0,
 XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_ABS = 0x1,
 XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_U8_BIAS = 0x2,
 XMEDIA_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8 = 0x3,
 XMEDIA_IVE_16BIT_TO_8BIT_MODE_BUTT
}xmedia_ive_16bit_to_8bit_mode_e;
```

#### 【成员】

| 成员名称                                         | 描述                              |
|----------------------------------------------|---------------------------------|
| XMEDIA_IVE_16BIT_TO_8BIT_MODE_S16_TO_S8      | S16 数据到 S8 数据的线性变换。             |
| XMEDIA_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。 |
| XMEDIA_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。 |
| XMEDIA_IVE_16BIT_TO_8BIT_MODE_U16_TO_U8      | U16 数据线性变换到 U8 数据。              |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_16bit\\_to\\_8bit](#) 中的【注意】，示意图请参见图 2-11。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_16bit\\_to\\_8bit\\_ctrl\\_s](#)

[xmedia\\_ive\\_16bit\\_to\\_8bit\\_ctrl\\_s](#)

#### 【说明】

定义 16bit 图像数据到 8bit 图像数据的转化控制参数。

#### 【定义】

```
typedef struct xmedia_ive_16bit_to_8bit_ctrl_s
{
 xmedia_ive_16bit_to_8bit_mode_e en_mode;
 xmedia_u16 u16_denominator;
 xmedia_u8 u8_numerator;
 xmedia_s8 s8_bias;
}xmedia_ive_16bit_to_8bit_ctrl_s;
```

#### 【成员】

| 成员名称            | 描述                                              |
|-----------------|-------------------------------------------------|
| en_mode         | 16bit 数据到 8bit 数据的转换模式。                         |
| u16_denominator | 线性变换中的分母。<br>取值范围: [max{1,u8_numerator}, 65535] |
| u8_numerator    | 线性变换中的分子。<br>取值范围: [0,255]。                     |
| s8_bias         | 线性变换中的平移项。<br>取值范围: [-128,127]。                 |

#### 【注意事项】

- 计算公式请参见 [xmedia\\_ive\\_thresh\\_u16](#) 中的【注意】，示意图请参见图 2-10。
- $u8\_numerator \leq u16\_denominator$ ，且  $u16\_denominator \neq 0$ ；

#### 【相关数据类型及接口】

[xmedia\\_ive\\_16bit\\_to\\_8bit\\_mode\\_e](#)

[xmedia\\_ive\\_ord\\_stat\\_filter\\_mode\\_e](#)

#### 【说明】

定义顺序统计量滤波模式。

#### 【定义】

```
typedef enum xmedia_ive_ord_stat_filter_mode_e
{
 XMEDIA_IVE_ORD_STAT_FILTER_MODE_MEDIAN = 0x0,
 XMEDIA_IVE_ORD_STAT_FILTER_MODE_MAX = 0x1,
 XMEDIA_IVE_ORD_STAT_FILTER_MODE_MIN = 0x2,
 XMEDIA_IVE_ORD_STAT_FILTER_MODE_BUTT
}xmedia_ive_ord_stat_filter_mode_e;
```

#### 【成员】

| 成员名称                                   | 描述    |
|----------------------------------------|-------|
| XMEDIA_IVE_ORD_STAT_FILTER_MODE_MEDIAN | 中值滤波。 |

| 成员名称                                | 描述               |
|-------------------------------------|------------------|
| XMEDIA_IVE_ORD_STAT_FILTER_MODE_MAX | 最大值滤波，等价于灰度图的膨胀。 |
| XMEDIA_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】

[xmedia\\_ive\\_ord\\_stat\\_filter\\_ctrl\\_s](#)

xmedia\_ive\_ord\_stat\_filter\_ctrl\_s

#### 【说明】

定义顺序统计量滤波控制参数。

#### 【定义】

```
typedef struct xmedia_ive_ord_stat_filter_ctrl_s
{
 xmedia_ive_ord_stat_filter_mode_e en_mode;
}xmedia_ive_ord_stat_filter_ctrl_s;
```

#### 【成员】

| 成员名称    | 描述         |
|---------|------------|
| en_mode | 顺序统计量滤波模式。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_ord\\_stat\\_filter\\_mode\\_e](#)

## xmedia\_ive\_map\_lut\_mem\_s

### 【说明】

定义 Map 算子的查找表内存信息。

### 【定义】

```
typedef struct xmedia_ive_map_lut_mem_s
{
 xmedia_u8 au8_map[XMEDIA_IVE_MAP_NUM];
}xmedia_ive_map_lut_mem_s;
```

### 【成员】

| 成员名称                        | 描述         |
|-----------------------------|------------|
| au8_map[XMEDIA_IVE_MAP_NUM] | Map 查找表数组。 |

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## xmedia\_ive\_map\_u8bit\_lut\_mem\_s

### 【说明】

定义 Map U8C1→U8C1 的查找表内存。

### 【定义】

```
typedef struct xmedia_ive_map_u8bit_lut_mem_s
{
 xmedia_u8 au8_map[XMEDIA_IVE_MAP_NUM];
}xmedia_ive_map_u8bit_lut_mem_s;
```

### 【成员】

| 成员名称                        | 描述         |
|-----------------------------|------------|
| au8_map[XMEDIA_IVE_MAP_NUM] | Map 查找表数组。 |

【注意事项】

无。

【相关数据类型及接口】

无。

## xmedia\_ive\_map\_u16bit\_lut\_mem\_s

【说明】

定义 Map U8C1→U16C1 的查找表内存。

【定义】

```
typedef struct xmedia_ive_map_u16bit_lut_mem_s
{
 xmedia_u16 au16_map[XMEDIA_IVE_MAP_NUM];
}xmedia_ive_map_u16bit_lut_mem_s;
```

【成员】

| 成员名称                         | 描述         |
|------------------------------|------------|
| au16_map[XMEDIA_IVE_MAP_NUM] | Map 查找表数组。 |

【注意事项】

无。

【相关数据类型及接口】

无。

## xmedia\_ive\_map\_s16bit\_lut\_mem\_s

【说明】

定义 Map U8C1→S16C1 的查找表内存。

【定义】

```
typedef struct xmedia_ive_map_s16bit_lut_mem_s
{
 xmedia_s16 as16_map[XMEDIA_IVE_MAP_NUM];
}xmedia_ive_map_s16bit_lut_mem_s;
```

#### 【成员】

| 成员名称                         | 描述         |
|------------------------------|------------|
| as16_map[XMEDIA_IVE_MAP_NUM] | Map 查找表数组。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_map\_mode\_e

#### 【说明】

定义 Map 的模式。

#### 【定义】

```
typedef enum xmedia_ive_map_mode_e
{
 XMEDIA_IVE_MAP_MODE_U8 = 0x0,
 XMEDIA_IVE_MAP_MODE_S16 = 0x1,
 XMEDIA_IVE_MAP_MODE_U16 = 0x2,
 XMEDIA_IVE_MAP_MODE_BUTT
}xmedia_ive_map_mode_e;
```

#### 【成员】

| 成员名称                    | 描述                 |
|-------------------------|--------------------|
| XMEDIA_IVE_MAP_MODE_U8  | U8C1→U8C1 Map 模式。  |
| XMEDIA_IVE_MAP_MODE_S16 | U8C1→U16C1 Map 模式。 |
| XMEDIA_IVE_MAP_MODE_U16 | U8C1→S16C1 Map 模式  |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

## xmedia\_ive\_map\_ctrl\_s

### 【说明】

定义 Map 控制参数。

### 【定义】

```
typedef struct xmedia_ive_map_ctrl_s
{
 xmedia_ive_map_mode_e en_mode;
}xmedia_ive_map_ctrl_s;
```

### 【成员】

| 成员名称    | 描述      |
|---------|---------|
| en_mode | Map 模式。 |

### 【注意事项】

无。

### 【相关数据类型及接口】

无。

## xmedia\_ive\_equalize\_hist\_ctrl\_mem\_s

### 【说明】

定义直方图均衡化辅助内存。

### 【定义】

```
typedef struct xmedia_ive_equalize_hist_ctrl_mem_s
{
 xmedia_u32 au32_hist[XMEDIA_IVE_HIST_NUM];
 xmedia_u8 au8Map[XMEDIA_IVE_MAP_NUM];
}xmedia_ive_equalize_hist_ctrl_mem_s;
```

### 【成员】

| 成员名称                           | 描述                    |
|--------------------------------|-----------------------|
| au32_hist[XMEDIA_IVE_HIST_NUM] | 直方图统计的输出。             |
| au8_map[XMEDIA_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】

[xmedia\\_ive\\_equalize\\_hist\\_ctrl\\_s](#)

### xmedia\_ive\_equalize\_hist\_ctrl\_s

#### 【说明】

定义直方图均衡化控制参数。

#### 【定义】

```
typedef struct xmedia_ive_equalize_hist_ctrl_s
{
 xmedia_ive_mem_info_s stmem;
}xmedia_ive_equalize_hist_ctrl_s;
```

#### 【成员】

| 成员名称  | 描述                                                                        |
|-------|---------------------------------------------------------------------------|
| stmem | 需开辟 sizeof( <a href="#">xmedia_ive_equalize_hist_ctrl_mem_s</a> )字节大小的内存。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_equalize\\_hist\\_ctrl\\_mem\\_s](#)

### xmedia\_ive\_add\_ctrl\_s

#### 【说明】

定义两图像的加权加控制参数。

#### 【定义】

```
typedef struct xmedia_ive_add_ctrl_s
{
 xmedia_u0q16 u0q16x; /*x of "xA+yB"*/
 xmedia_u0q16 u0q16y; /*y of "xA+yB"*/
}xmedia_ive_add_ctrl_s;
```

#### 【成员】

| 成员名称   | 描述                     |
|--------|------------------------|
| u0q16x | 加权加 "xA+yB" 中的权重 "x" 。 |
| u0q16y | 加权加 "xA+yB" 中的权重 "y" 。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

xmedia\_ive\_ncc\_dst\_mem\_s

#### 【说明】

定义 NCC 的输出内存信息。

#### 【定义】

```
typedef struct xmedia_ive_ncc_dst_mem_s
{
 xmedia_u64 u64_numerator;
 xmedia_u64 u64_quad_sum1;
 xmedia_u64 u64_quad_sum2;
 xmedia_u8 u8_reserved[8];
}xmedia_ive_ncc_dst_mem_s;
```

#### 【成员】

| 成员名称          | 描述                                                                             |
|---------------|--------------------------------------------------------------------------------|
| u64_numerator | NCC 计算公式的分子--<br>$\sum_{i=1}^w \sum_{j=1}^h (I_{src1}(i, j) * I_{src2}(i, j))$ |
| u64_quad_sum1 | NCC 计算公式的分母--根号内部分:<br>$\sum_{i=1}^w \sum_{j=1}^h (I_{src1}^2(i, j))。$         |
| u64_quad_sum2 | NCC 计算公式的分母--根号内部分:<br>$\sum_{i=1}^w \sum_{j=1}^h (I_{src2}^2(i, j))$          |
| u8_reserved   | 保留字段。                                                                          |

#### 【注意事项】

计算公式请参见 [xmedia\\_ive\\_ncc](#) 中的【注意】。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_region\_s

#### 【说明】

定义连通区域信息。

#### 【定义】

```
typedef struct xmedia_ive_region_s
{
 xmedia_u32 u32_area; /*Represented by the pixel number*/
 xmedia_u16 u16_left; /*Circumscribed rectangle left border*/
 xmedia_u16 u16_right; /*Circumscribed rectangle right border*/
 xmedia_u16 u16_top; /*Circumscribed rectangle top border*/
 xmedia_u16 u16_bottom; /*Circumscribed rectangle bottom border*/
}xmedia_ive_region_s;
```

#### 【成员】

| 成员名称       | 描述                  |
|------------|---------------------|
| u32_area   | 连通区域面积，以连通区域像素数目表示。 |
| u16_left   | 连通区域外接矩形的最左边坐标。     |
| u16_right  | 连通区域外接矩形的最右边坐标。     |
| u16_top    | 连通区域外接矩形的最上面坐标。     |
| u16_bottom | 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】

[xmedia\\_ive\\_ccblob\\_s](#)

xmedia\_ive\_ccblob\_s

#### 【说明】

定义连通区域标记的输出信息。

#### 【定义】

```
typedef struct xmedia_ive_ccblob_s
{
 xmedia_u16 u16_cur_area_thr; /*Threshold of the result regions' area*/
 xmedia_s8 s8_label_status; /*-1: Labeled failed ; 0: Labeled successfully*/
 xmedia_u8 u8_region_num; /*Number of valid region, non-continuous stored*/
 xmedia_ive_region_s ast_region[XMEDIA_IVE_MAX_REGION_NUM]; /*Valid regions with
'u32_area>0' and 'label = ArrayIndex+1'*/
}xmedia_ive_ccblob_s;
```

#### 【成员】

| 成员名称             | 描述                                           |
|------------------|----------------------------------------------|
| u16_cur_area_thr | 有效连通区域的面积阈值，<br>ast_region 中面积小于这个阈值的都被置为 0。 |

| 成员名称                                  | 描述                                  |
|---------------------------------------|-------------------------------------|
| s8_label_status                       | 连通区域标记是否成功。<br>-1：标记失败；<br>0：标记成功。  |
| u8_region_num                         | 有效连通区域个数。                           |
| ast_region[XMEDIA_IVE_MAX_REGION_NUM] | 连通区域信息：有效的连通区域其面积大于 0，对应标记为数组下标加 1。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_region\\_s](#)

[xmedia\\_ive\\_ccl\\_mode\\_e](#)

#### 【说明】

定义连通区域模式。

#### 【定义】

```
typedef enum xmedia_ive_ccl_mode_e
{
 XMEDIA_IVE_CCL_MODE_4C = 0x0, /*4-connectivity*/
 XMEDIA_IVE_CCL_MODE_8C = 0x1, /*8-connectivity*/
 XMEDIA_IVE_CCL_MODE_BUTT
}xmedia_ive_ccl_mode_e;
```

#### 【成员】

| 成员名称                   | 描述    |
|------------------------|-------|
| XMEDIA_IVE_CCL_MODE_4C | 4-连通。 |
| XMEDIA_IVE_CCL_MODE_8C | 8-连通。 |

【注意事项】

无。

【相关数据类型及接口】

无。

## xmedia\_ive\_ccl\_ctrl\_s

【说明】

定义连通区域标记控制参数。

【定义】

```
typedef struct xmedia_ive_ccl_ctrl_s
{
 xmedia_ive_ccl_mode_e en_mode; /*Mode*/
 xmedia_u16 u16_init_area_thr; /*Init threshold of region area*/
 xmedia_u16 u16_step; /*Increase area step for once*/
}xmedia_ive_ccl_ctrl_s;
```

【成员】

| 成员名称              | 描述                                      |
|-------------------|-----------------------------------------|
| en_mode           | 连通区域模式。                                 |
| u16_init_area_thr | 初始面积阈值。<br>取值范围：[0, 65535]。<br>参考取值：4。  |
| u16_step          | 面积阈值增长步长。<br>取值范围：[1,65535]。<br>参考取值：2。 |

【注意事项】

无。

【相关数据类型及接口】

[xmedia\\_ive\\_ccblob\\_s](#)

## xmedia\_ive\_gmm\_ctrl\_s

### 【说明】

定义 GMM 背景建模的控制参数。

### 【定义】

```
typedef struct xmedia_ive_gmm_ctrl_s
{
 xmedia_u22q10 u22q10_noise_var; /*Initial noise Variance*/
 xmedia_u22q10 u22q10_max_var; /*Max Variance*/
 xmedia_u22q10 u22q10_min_var; /*Min Variance*/
 xmedia_u0q16 u0q16_learn_rate; /*Learning rate*/
 xmedia_u0q16 u0q16_bg_ratio; /*Background ratio*/
 xmedia_u8q8 u8q8_var_thr; /*Variance Threshold*/
 xmedia_u0q16 u0q16_init_weight; /*Initial Weight*/
 xmedia_u8 u8_model_num; /*Model number: 3 or 5*/
}xmedia_ive_gmm_ctrl_s;
```

### 【成员】

| 成员名称             | 描述                                                                                                                                                                                                                 |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| u22q10_noise_var | 初始噪声方差。<br>取值范围：[0x1, 0xFFFFFFFF]。<br>对灰度的 GMM，对应 OpenCV MOG 中灰度模型中的 noiseSigma * noiseSigma。<br>参考取值：15*15*(1<<10)。<br>对 RGB 的 GMM，对应 OpenCV MOG 中 RGB 模型中的 3 * noiseSigma * noiseSigma。<br>参考取值：3*15*15*(1<<10)。 |
| u22q10_max_var   | 模型方差的最大值。<br>取值范围；[0x1, 0xFFFFFFFF]。<br>对应 OpenCV MOG2 中 fVarMax。<br>参考取值：3*4000<<10 (RGB)，2000<<10（灰度）。                                                                                                           |

| 成员名称              | 描述                                                                                                                                          |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| u22q10_min_var    | 模型方差的最小值。<br>取值范围：[0x1, u22q10_max_var]。<br>对应 OpenCV MOG2 中 fVarMin。<br>参考取值：600<<10 (RGB), 200<<10 (灰度)。                                  |
| u0q16_learn_rate  | 学习速率。<br>取值范围：[1, 65535]。<br>对应 OpenCV MOG2 中 learningRate。<br>参考取值：if (frameNum<500) (1/frameNum)*((1<<16)-1); else ((1/500)*((1<<16)-1))。 |
| u0q16_bg_ratio    | 背景比例阈值。<br>取值范围：[1, 65535]。<br>对应 OpenCV MOG 中 backgroundRatio。<br>参考取值：0.8*((1<<16)-1)。                                                    |
| u8q8_var_thr      | 方差阈值。<br>取值范围：[1, 65535]。<br>对应 OpenCV MOG 中 varThreshold，用于决定一个像素是否命中当前模型。<br>参考取值：6.25*(1<<8)。                                            |
| u0q16_init_weight | 初始权重。<br>取值范围：[1, 65535]。<br>对应 OpenCV MOG 中的 defaultInitialWeight。<br>参考取值：0.05*((1<<16)-1)。                                               |
| u8_model_num      | 模型个数。<br>取值范围：{3,5}。<br>对应 OpenCV MOG 中 nmixtures。                                                                                          |

【注意事项】

无。

【相关数据类型及接口】

无

xmedia\_ive\_gmm2\_sns\_factor\_mode\_e

【说明】

定义灵敏度系数模式。

【定义】

```
typedef enum xmedia_ive_gmm2_sns_factor_mode_e
{
 XMEDIA_IVE_GMM2_SNS_FACTOR_MODE_GLB = 0x0, /*Global sensitivity factor
mode*/
 XMEDIA_IVE_GMM2_SNS_FACTOR_MODE_PIX = 0x1, /*Pixel sensitivity factor
mode*/
 XMEDIA_IVE_GMM2_SNS_FACTOR_MODE_BUTT
}xmedia_ive_gmm2_sns_factor_mode_e;
```

【成员】

| 成员名称                                | 描述                                                                                         |
|-------------------------------------|--------------------------------------------------------------------------------------------|
| XMEDIA_IVE_GMM2_SNS_FACTOR_MODE_GLB | 全局灵敏度系数模式，每个像素在模型匹配过程中，方差灵敏度使用 <a href="#">xmedia_ive_gmm2_ctrl_s</a> 的 u8_glb_sns_factor。 |
| XMEDIA_IVE_GMM2_SNS_FACTOR_MODE_PIX | 像素级灵敏度系数模式，每个像素在模型匹配过程中，方差灵敏度使用 pstFactor 的灵敏度系数。                                          |

【注意事项】

无。

【相关数据类型及接口】

无。

xmedia\_ive\_gmm2\_life\_update\_factor\_mode\_e

【说明】

定义模型时长参数更新模式。

【定义】

```
typedef enum xmedia_ive_gmm2_life_update_factor_mode_e
{
 XMEDIA_IVE_GMM2_LIFE_UPDATE_FACTOR_MODE_GLB = 0x0, /*Global life update
factor mode*/
 XMEDIA_IVE_GMM2_LIFE_UPDATE_FACTOR_MODE_PIX = 0x1, /*Pixel life update
factor mode*/
 XMEDIA_IVE_GMM2_LIFE_UPDATE_FACTOR_MODE_BUTT
}xmedia_ive_gmm2_life_update_factor_mode_e ;
```

【成员】

| 成员名称                                        | 描述                                                                                                 |
|---------------------------------------------|----------------------------------------------------------------------------------------------------|
| XMEDIA_IVE_GMM2_LIFE_UPDATE_FACTOR_MODE_GLB | 模型时长参数全局更新模式，每个像素模型时长参数在更新时使用 <a href="#">xmedia_ive_gmm2_ctrl_s</a> 的 u16_glb_life_update_factor。 |
| XMEDIA_IVE_GMM2_LIFE_UPDATE_FACTOR_MODE_PIX | 模型时长参数像素级更新模式，每个像素模型时长在更新时使用 pstFactor 的模型更新参数。                                                    |

【注意事项】

无。

【相关数据类型及接口】

无。

xmedia\_ive\_gmm2\_ctrl\_s

【说明】

定义 GMM2 背景建模的控制参数。

【定义】

```

typedef struct xmedia_ive_gmm2_ctrl_s
{
 xmedia_ive_gmm2_sns_factor_mode_e en_sns_factor_mode; /*Sensitivity factor
mode*/
 xmedia_ive_gmm2_life_update_factor_mode_e en_life_update_factor_mode; /*Life
update factor mode*/
 xmedia_u16 u16_glb_life_update_factor; /*Global life update factor (default: 4)*/
 xmedia_u16 u16_life_thr; /*Life threshold (default: 5000)*/
 xmedia_u16 u16_freq_init_val; /*Initial frequency (default: 20000)*/
 xmedia_u16 u16_freq_redu_factor; /*Frequency reduction factor (default: 0xFF00)*/
 xmedia_u16 u16_freq_add_factor; /*Frequency adding factor (default: 0xEF)*/
 xmedia_u16 u16_freq_thr; /*Frequency threshold (default: 12000)*/
 xmedia_u16 u16_var_rate; /*Variation update rate (default: 1)*/
 xmedia_u9q7 u9q7_max_var; /*Max variation (default: (16 * 16)<<7)*/
 xmedia_u9q7 u9q7_min_var; /*Min variation (default: (8 * 8)<<7)*/
 xmedia_u8 u8_glb_sns_factor; /*Global sensitivity factor (default: 8)*/
 xmedia_u8 u8_model_num; /*Model number (range: 1~5, default: 3)*/
}xmedia_ive_gmm2_ctrl_s;

```

#### 【成员】

| 成员名称                       | 描述                                                                                                                           |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------|
| en_sns_factor_mode         | 灵敏度模式，默认全局模式。全局模式使用 u8_glb_sns_factor 作为灵敏度系数；像素模式使用 xmedia_ive_gmm2 中 pstFactor 的低 8bit 值作为灵敏度系数。                           |
| en_life_update_factor_mode | 模型时长更新模式，默认全局模式。全局模式使用 u16_glb_life_update_factor 作为前进模型更新参数；像素模式使用 xmedia_ive_gmm2 中 pst 中 pstFactor 的高 8bit 值作为前进模型时长更新参数。 |
| u16_glb_life_update_factor | 全局模型更新参数，范围：0-65535，默认：4。                                                                                                    |
| u16_life_thr               | 背景模型生成时间，表示一个模型从前景模型转成背景模型需要的时间，范围：0-65535，默认：5000。                                                                          |
| u16_freq_init_val          | 初始频率，范围：0-65535，默认：20000。                                                                                                    |
| u16_freq_redu_factor       | 频率衰减系数，范围：0-65535，默认：0xFF00。                                                                                                 |

| 成员名称                | 描述                              |
|---------------------|---------------------------------|
| u16_freq_add_factor | 模型匹配频率增加系数，范围：0-65535，默认：0xEF。  |
| u16_freq_thr        | 模型失效频率阈值，范围：0-65535，默认：12000。   |
| u16_var_rate        | 方差更新率，范围：0-65535，默认：1。          |
| u9q7_max_var        | 方差最大值，范围：0-65535，默认：(16x16)<<7。 |
| u9q7_min_var        | 方差最小值，范围：0-方差最大值，默认：(8x8)<<7。   |
| u8_glb_sns_factor   | 全局灵敏度参数，范围：0-255，默认：8。          |
| u8_model_num        | 模型数量，范围 1-5，默认：3。               |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_canny\_stack\_size\_s

#### 【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 xmedia_ive_canny_stack_size_s
{
 xmedia_u32 u32StackSize; /*Stack size for output*/
 xmedia_u8 u8_reserved[12]; /*For 16 byte align*/
}xmedia_ive_canny_stack_size_s;
```

#### 【成员】

| 成员名称            | 描述            |
|-----------------|---------------|
| u32StackSize    | 栈大小(强边缘点的个数)。 |
| u8_reserved[12] | 保留位。          |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_canny\_hys\_edge\_ctrl\_s

#### 【说明】

定义 Canny 边缘前半部分计算任务的控制参数。

#### 【定义】

```
typedef struct xmedia_ive_canny_hys_edge_ctrl_s
{
 xmedia_ive_mem_info_s stmem;
 xmedia_u16 u16_low_thr;
 xmedia_u16 u16_high_thr;
 xmedia_s8 as8_mask[25];
} xmedia_ive_canny_hys_edge_ctrl_s;
```

#### 【成员】

| 成员名称         | 描述                              |
|--------------|---------------------------------|
| stmem        | 辅助内存。                           |
| u16_low_thr  | 低阈值。<br>取值范围：[0,255]。           |
| u16_high_thr | 高阈值。<br>取值范围：[u16_low_thr,255]。 |
| as8_mask[25] | 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。

## xmedia\_ive\_lbp\_cmp\_mode\_e

【说明】

定义 LBP 计算的比较模式。

【定义】

```
typedef enum xmedia_ive_lbp_cmp_mode_e
{
 XMEDIA_IVE_LBP_CMP_MODE_NORMAL = 0x0, /* P(x)-P(center)>= un8_bit_thr.s8Val,
s(x)=1; else s(x)=0; */
 XMEDIA_IVE_LBP_CMP_MODE_ABS = 0x1, /* abs(P(x)-
P(center))>=un8_bit_thr.u8Val, s(x)=1; else s(x)=0; */
 XMEDIA_IVE_LBP_CMP_MODE_BUTT
}xmedia_ive_lbp_cmp_mode_e;
```

【成员】

| 成员名称                           | 描述           |
|--------------------------------|--------------|
| XMEDIA_IVE_LBP_CMP_MODE_NORMAL | LBP 简单比较模式。  |
| XMEDIA_IVE_LBP_CMP_MODE_ABS    | LBP 绝对值比较模式。 |

【注意事项】

计算公式参考 [xmedia\\_ive\\_lbp](#) 中的【注意】，示意图请参考图 2-18。

【相关数据类型及接口】

[xmedia\\_ive\\_lbp\\_ctrl\\_s](#)

## xmedia\_ive\_lbp\_ctrl\_s

【说明】

定义 LBP 纹理计算控制参数。

### 【定义】

```
typedef struct xmedia_ive_lbp_ctrl_s
{
 xmedia_ive_lbp_cmp_mode_e en_mode;
 xmedia_ive_8bit_u un8_bit_thr;
}xmedia_ive_lbp_ctrl_s;
```

### 【成员】

| 成员名称        | 描述                                                                                                            |
|-------------|---------------------------------------------------------------------------------------------------------------|
| en_mode     | LBP 比较模式。                                                                                                     |
| un8_bit_thr | LBP 比较阈值。<br>XMEDIA_IVE_LBP_CMP_MODE_NORMAL 下的取值范围：[-128,127]。<br>XMEDIA_IVE_LBP_CMP_MODE_ABS 下的取值范围：[0,255]。 |

### 【注意事项】

计算公式参考 [xmedia\\_ive\\_lbp](#) 中的【注意】，示意图请参考图 2-18。

### 【相关数据类型及接口】

[xmedia\\_ive\\_lbp\\_cmp\\_mode\\_e](#)  
[xmedia\\_ive\\_8bit\\_u](#)

[xmedia\\_ive\\_norm\\_grad\\_out\\_ctrl\\_e](#)

### 【说明】

定义归一化梯度信息计算任务输出控制枚举类型。

### 【定义】

```
typedef enum xmedia_ive_norm_grad_out_ctrl_e
{
 XMEDIA_IVE_NORM_GRAD_OUT_CTRL_HOR_AND_VER = 0x0,
 XMEDIA_IVE_NORM_GRAD_OUT_CTRL_HOR = 0x1,
 XMEDIA_IVE_NORM_GRAD_OUT_CTRL_VER = 0x2,
 XMEDIA_IVE_NORM_GRAD_OUT_CTRL_COMBINE = 0x3,
 XMEDIA_IVE_NORM_GRAD_OUT_CTRL_BUTT
}xmedia_ive_norm_grad_out_ctrl_e;
```

### 【成员】

| 成员名称                                      | 描述                                                                    |
|-------------------------------------------|-----------------------------------------------------------------------|
| XMEDIA_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 <a href="#">xmedia_ive_normgrad</a> 的【参数】)。 |
| XMEDIA_IVE_NORM_GRAD_OUT_CTRL_HOR         | 仅输出梯度信息的 H 分量图。                                                       |
| XMEDIA_IVE_NORM_GRAD_OUT_CTRL_VER         | 仅输出梯度信息的 V 分量图。                                                       |
| XMEDIA_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】

[xmedia\\_ive\\_norm\\_grad\\_ctrl\\_s](#)

xmedia\_ive\_norm\_grad\_ctrl\_s

### 【说明】

定义归一化梯度信息计算控制参数。

### 【定义】

```
typedef struct xmedia_ive_norm_grad_ctrl_s
{
 xmedia_ive_norm_grad_out_ctrl_e en_out_ctrl;
 xmedia_s8 as8_mask[25];
 xmedia_u8 u8_norm;
}xmedia_ive_norm_grad_ctrl_s;
```

### 【成员】

| 成员名称         | 描述          |
|--------------|-------------|
| en_out_ctrl  | 梯度信息输出控制模式。 |
| as8_mask[25] | 计算梯度需要的模板。  |

| 成员名称    | 描述                     |
|---------|------------------------|
| u8_norm | 归一化参数。<br>取值范围：[1,13]。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_norm\\_grad\\_out\\_ctrl\\_e](#)

[xmedia\\_ive\\_lk\\_optical\\_flow\\_pyr\\_out\\_mode\\_e](#)

#### 【说明】

定义金字塔 LK 光流计算输出模式。

#### 【定义】

```
typedef enum xmedia_ive_lk_optical_flow_pyr_out_mode_e
{
 XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_NONE = 0, /*Output none*/
 XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_STATUS = 1, /*Output status*/
 XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_BOTH = 2, /*Output status and
err*/
 XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_BUTT
}xmedia_ive_lk_optical_flow_pyr_out_mode_e;
```

#### 【成员】

| 成员名称                                           | 描述                        |
|------------------------------------------------|---------------------------|
| XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_NONE   | pstStatus 以及 pstErr 均不输出。 |
| XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_STATUS | 仅输出 pstStatus。            |
| XMEDIA_IVE_LK_OPTICAL_FLOW_PYR_OUT_MODE_BOTH   | 同时输出 pstStatus 和 pstErr。  |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_lk\_optical\_flow\_pyr\_ctrl\_s

#### 【说明】

定义金字塔 LK 光流计算控制参数。

#### 【定义】

```
typedef struct xmedia_ive_lk_optical_flow_pyr_ctrl_s
{
 xmedia_ive_lk_optical_flow_pyr_out_mode_e en_out_mode;
 xmedia_bool b_use_init_flow; /*where to use initial flow*/
 xmedia_u16 u16_pts_num; /*Number of the feature points,<=500*/
 xmedia_u8 u8_max_level; /*0<=u8_max_level<=3*/
 xmedia_u0q8 u0q8_min_eigthr; /*Minimum eigenvalue threshold*/
 xmedia_u8 u8_iter_cnt; /*Maximum iteration times, <=20*/
 xmedia_u0q8 u0q8_eps; /*Used for exit criteria: dx^2 + dy^2 < u0q8_eps */
}xmedia_ive_lk_optical_flow_pyr_ctrl_s;
```

#### 【成员】

| 成员名称            | 描述                                                                               |
|-----------------|----------------------------------------------------------------------------------|
| en_out_mode     | pstStatus 以及 pstErr 的输出模式控制。                                                     |
| b_use_init_flow | 是否使用初始光流计算（pstNextPts 是否需要初始化）：<br>XMEDIA_TRUE 表示使用初始光流，XMEDIA_FALSE 表示不适用初始光流。  |
| u16_pts_num     | pstPrevPts、pstNextPts 中特征点个数，同时也是<br>pstStatus 和 pstErr 的数组大小。<br>取值范围：[1, 500]。 |

| 成员名称            | 描述                                                                   |
|-----------------|----------------------------------------------------------------------|
| u8_max_level    | u8_max_level+1 为金字塔层数相关。<br>取值范围：[0, 3]，对应金字塔层数[1, 4]。<br>参考取值：2。    |
| u0q8_min_eigthr | 最小特征值阈值。<br>取值范围：[1,255]。                                            |
| u8_iter_cnt     | 最大迭代次数。<br>取值范围：[1,20]。                                              |
| u0q8_eps        | 迭代收敛条件： $dx^2 + dy^2 < u0q8\_epsilon$ 。<br>取值范围：[1, 255]。<br>参考取值：2。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_st\_max\_eig\_s

#### 【说明】

定义 Shi-Tomas-like 角点计算时最大角点响应值结构体。

#### 【定义】

```
typedef struct xmedia_ive_st_max_eig_s
{
 xmedia_u16 u16_max_eig; /*Shi-Tomasi second step output MaxEig*/
 xmedia_u8 u8_reserved[14]; /*For 16 byte align*/
}xmedia_ive_st_max_eig_s;
```

#### 【成员】

| 成员名称            | 描述       |
|-----------------|----------|
| u16_max_eig     | 最大角点响应值。 |
| u8_reserved[14] | 保留位。     |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_st\_candi\_corner\_ctrl\_s

#### 【说明】

定义 Shi-Tomas-like 候选角点计算控制参数。

#### 【定义】

```
typedef struct xmedia_ive_st_candi_corner_ctrl_s
{
 xmedia_ive_mem_info_s stmem;
 xmedia_u0Q8 u0q8_quality_level;
}xmedia_ive_st_candi_corner_ctrl_s;
```

#### 【成员】

| 成员名称               | 描述                                                                                                |
|--------------------|---------------------------------------------------------------------------------------------------|
| stmem              | 辅助内存。内存配置大小见 <a href="#">xmedia_ive_st_candi_corner</a> 的【注意】。                                    |
| u0q8_quality_level | ShiTomas 角点质量控制参数，角点响应值小于“u0q8_quality_level * 最大角点响应值”的点将直接被确认为非角点。<br>取值范围：[1,255]。<br>参考取值：25。 |

#### 【注意事项】

无。

【相关数据类型及接口】

无。

## xmedia\_ive\_st\_corner\_info\_s

【说明】

定义 Shi-Tomas-like 角点计算输出的角点信息结构体。

【定义】

```
typedef struct xmedia_ive_st_corner_info_s
{
 xmedia_u16 u16_corner_num;
 xmedia_ive_point_u16_s ast_corner[XMEDIA_IVE_ST_MAX_CORNER_NUM];
}xmedia_ive_st_corner_info_s;
```

【成员】

| 成员名称                                     | 描述      |
|------------------------------------------|---------|
| u16_corner_num                           | 有效角点数目。 |
| ast_corner[XMEDIA_IVE_ST_MAX_CORNER_NUM] | 角点坐标数组。 |

【注意事项】

无。

【相关数据类型及接口】

无。

## xmedia\_ive\_st\_corner\_ctrl\_s

【说明】

定义 Shi-Tomas-like 角点筛选控制参数。

【定义】

```
typedef struct xmedia_ive_st_corner_ctrl_s
{
 xmedia_u16 u16_max_corner_num;
```

```
xmedia_u16 u16_min_dist;
}xmedia_ive_st_corner_ctrl_s;
```

#### 【成员】

| 成员名称               | 描述                                     |
|--------------------|----------------------------------------|
| u16_max_corner_num | 最大角点数目。<br>取值范围：[1,500]。               |
| u16_min_dist       | 相邻角点最小距离。<br>取值范围：[1,255]。<br>参考取值：10。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

xmedia\_ive\_sad\_mode\_e

#### 【说明】

定义 SAD 计算模式。

#### 【定义】

```
typedef enum xmedia_ive_sad_mode_e
{
 XMEDIA_IVE_SAD_MODE_MB_4X4 = 0x0, /*4x4*/
 XMEDIA_IVE_SAD_MODE_MB_8X8 = 0x1, /*8x8*/
 XMEDIA_IVE_SAD_MODE_MB_16X16 = 0x2, /*16x16*/
 XMEDIA_IVE_SAD_MODE_BUTT
}xmedia_ive_sad_mode_e;
```

#### 【成员】

| 成员名称                       | 描述               |
|----------------------------|------------------|
| XMEDIA_IVE_SAD_MODE_MB_4X4 | 按 4x4 像素块计算 SAD。 |

| 成员名称                         | 描述                 |
|------------------------------|--------------------|
| XMEDIA_IVE_SAD_MODE_MB_8X8   | 按 8x8 像素块计算 SAD。   |
| XMEDIA_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】

[xmedia\\_ive\\_sad\\_ctrl\\_s](#)

[xmedia\\_ive\\_sad\\_out\\_ctrl\\_e](#)

#### 【说明】

定义 SAD 输出控制模式。

#### 【定义】

```
typedef enum xmedia_ive_sad_out_ctrl_e
{
 XMEDIA_IVE_SAD_OUT_CTRL_16BIT_BOTH= 0x0, /*Output 16 bit sad and thresh*/
 XMEDIA_IVE_SAD_OUT_CTRL_8BIT_BOTH = 0x1, /*Output 8 bit sad and thresh*/
 XMEDIA_IVE_SAD_OUT_CTRL_16BIT_SAD = 0x2, /*Output 16 bit sad*/
 XMEDIA_IVE_SAD_OUT_CTRL_8BIT_SAD = 0x3, /*Output 8 bit sad*/
 XMEDIA_IVE_SAD_OUT_CTRL_THRESH = 0x4, /*Output thresh */
 XMEDIA_IVE_SAD_OUT_CTRL_BUTT
}xmedia_ive_sad_out_ctrl_e;
```

#### 【成员】

| 成员名称                               | 描述                     |
|------------------------------------|------------------------|
| XMEDIA_IVE_SAD_OUT_CTRL_16BIT_BOTH | 16 bit SAD 图和阈值化图输出模式。 |
| XMEDIA_IVE_SAD_OUT_CTRL_8BIT_BOTH  | 8 bit SAD 图和阈值化图输出模式。  |
| XMEDIA_IVE_SAD_OUT_CTRL_16BIT_SAD  | 16 bit SAD 图输出模式。      |
| XMEDIA_IVE_SAD_OUT_CTRL_8BIT_SAD   | 8 bit SAD 图输出模式。       |

| 成员名称                           | 描述        |
|--------------------------------|-----------|
| XMEDIA_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】

[xmedia\\_ive\\_sad\\_ctrl\\_s](#)

[xmedia\\_ive\\_sad\\_ctrl\\_s](#)

#### 【说明】

定义 SAD 控制参数。

#### 【定义】

```
typedef struct xmedia_ive_sad_ctrl_s
{
 xmedia_ive_sad_mode_e en_mode;
 xmedia_ive_sad_out_ctrl_e en_out_ctrl;
 xmedia_u16 u16_thr; /*srcVal <= u16_thr, dstVal = minVal; srcVal >
u16_thr, dstVal = maxVal.*/
 xmedia_u8 u8_min_val; /*Min value*/
 xmedia_u8 u8_max_val; /*Max value*/
}xmedia_ive_sad_ctrl_s;
```

#### 【成员】

| 成员名称        | 描述                                                                                                                                     |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------|
| en_mode     | SAD 计算模式。                                                                                                                              |
| en_out_ctrl | SAD 输出控制模式。                                                                                                                            |
| u16_thr     | 对计算的 SAD 图进行阈值化的阈值。<br>取值范围依赖 en_mode： <ul style="list-style-type: none"> <li>XMEDIA_IVE_SAD_OUT_CTRL_8BIT_BOTH, 取值[0, 255]</li> </ul> |

| 成员名称       | 描述                                                                                                                                   |
|------------|--------------------------------------------------------------------------------------------------------------------------------------|
|            | <ul style="list-style-type: none"> <li>XMEDIA_IVE_SAD_OUT_CTRL_16BIT_BOTH 和 XMEDIA_IVE_SAD_OUT_CTRL_THRESH, 取值 [0, 65535]</li> </ul> |
| u8_min_val | 阈值化不超过 u16_thr 时的取值。                                                                                                                 |
| u8_max_val | 阈值化超过 u16_thr 时的取值。                                                                                                                  |

#### 【注意事项】

无

#### 【相关数据类型及接口】

[xmedia\\_ive\\_sad\\_mode\\_e](#)

[xmedia\\_ive\\_sad\\_out\\_ctrl\\_e](#)

### xmedia\_ive\_grad\_fg\_mode\_e

#### 【说明】

定义梯度前景计算模式。

#### 【定义】

```
typedef enum xmedia_ive_grad_fg_mode_e
{
 XMEDIA_IVE_GRAD_FG_MODE_USE_CUR_GRAD = 0x0,
 XMEDIA_IVE_GRAD_FG_MODE_FIND_MIN_GRAD = 0x1,
 XMEDIA_IVE_GRAD_FG_MODE_BUTT
}xmedia_ive_grad_fg_mode_e;
```

#### 【成员】

| 成员名称                                  | 描述          |
|---------------------------------------|-------------|
| XMEDIA_IVE_GRAD_FG_MODE_USE_CUR_GRAD  | 当前位置梯度计算模式。 |
| XMEDIA_IVE_GRAD_FG_MODE_FIND_MIN_GRAD | 周边最小梯度计算模式。 |

#### 【注意事项】

无。

## 【相关数据类型及接口】

[xmedia\\_ive\\_grad\\_fg\\_ctrl\\_s](#)

xmedia\_ive\_grad\_fg\_ctrl\_s

## 【说明】

定义计算梯度前景控制参数。

## 【定义】

```
typedef struct xmedia_ive_grad_fg_ctrl_s
{
 xmedia_ive_grad_fg_mode_e en_mode; /*Calculation mode*/
 xmedia_u16 u16_ed_w_factor; /*Edge width adjustment factor (range: 500 to 2000;
 default: 1000)*/
 xmedia_u8 u8_crl_coef_thr; /*Gradient vector correlation coefficient
 threshold (ranges: 50 to 100; default: 80)*/
 xmedia_u8 u8_mag_crl_thr; /*Gradient amplitude threshold (range: 0 to 20;
 default: 4)*/
 xmedia_u8 u8_min_mag_diff; /*Gradient magnitude difference threshold
 (range: 2 to 8; default: 2)*/
 xmedia_u8 u8_noise_val; /*Gradient amplitude noise threshold (range: 1 to 8;
 default: 1)*/
 xmedia_u8 u8_ed_w_dark; /*Black pixels enable flag (range: 0 (no), 1 (yes);
 default: 1)*/
}xmedia_ive_grad_fg_ctrl_s;
```

## 【成员】

| 成员名称            | 描述                                                      |
|-----------------|---------------------------------------------------------|
| en_mode         | 梯度前景计算模式。参考 <a href="#">xmedia_ive_grad_fg_mode_e</a> 。 |
| u16_ed_w_factor | 边缘宽度调节因子。<br>取值范围：[500, 2000]。<br>参考取值：1000。            |
| u8_crl_coef_thr | 梯度向量相关系数阈值。<br>取值范围[50, 100]。<br>参考取值：80。               |

| 成员名称            | 描述                                  |
|-----------------|-------------------------------------|
| u8_mag_crl_thr  | 梯度幅值阈值。<br>取值范围[0, 20]。<br>参考取值：4。  |
| u8_min_mag_diff | 梯度幅值差值阈值。<br>取值范围[2, 8]。<br>参考取值：2。 |
| u8_noise_val    | 梯度幅值噪声阈值。<br>取值范围[1, 8]。<br>参考取值：1。 |
| u8_ed_w_dark    | 黑像素使能标志，0 表示不开启，1 表示开启。<br>参考取值：1。  |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_grad\\_fg\\_mode\\_e](#)

[xmedia\\_ive\\_candi\\_bg\\_pix\\_s](#)

#### 【说明】

定义候选背景模型数据。

#### 【定义】

```
typedef struct xmedia_ive_candi_bg_pix_s
{
 xmedia_u8Q4F4 u8q4f4_mean; /*Candidate background grays value */
 xmedia_u16 u16_start_time; /*Candidate Background start time */
 xmedia_u16 u16_sum_access_time; /*Candidate Background cumulative access time */
 xmedia_u16 u16_short_keep_time; /*Candidate background short hold time*/
 xmedia_u8 u8_chg_cond; /*Time condition for candidate background into the
changing state*/
 xmedia_u8 u8_poten_bg_life; /*Potential background cumulative access time */
}
```

```
}xmedia_ive_candi_bg_pix_s;
```

#### 【成员】

| 成员名称                | 描述                                  |
|---------------------|-------------------------------------|
| u8q4f4_mean         | 候选背景灰度均值，高 12 位为候选背景像素值，低 4 位为状态标识。 |
| u16_start_time      | 候选背景起始时间。                           |
| u16_sum_access_time | 候选背景累计访问时间。                         |
| u16_short_keep_time | 候选背景短时保持时间。                         |
| u8_chg_cond         | 候选背景进入变换状态时间条件。                     |
| u8_poten_bg_life    | 潜在背景生命值。                            |

#### 【注意事项】

无

#### 【相关数据类型及接口】

[xmedia\\_ive\\_work\\_bg\\_pix\\_s](#)

[xmedia\\_ive\\_bg\\_model\\_pix\\_s](#)

xmedia\_ive\_work\_bg\_pix\_s

#### 【说明】

定义工作背景模型数据。

#### 【定义】

```
typedef struct xmedia_ive_work_bg_pix_s
{
 xmedia_u8Q4F4 u8q4f4_mean; /*0# background grays value */
 xmedia_u16 u16_acc_time; /*Background cumulative access time */
 xmedia_u8 u8_pre_gray; /*Gray value of last pixel */
 xmedia_u5Q3 u5q3_diff_thr; /*Differential threshold */
 xmedia_u8 u8_acc_flag; /*Background access flag */
 xmedia_u8 u8_bg_gray[3]; /*1# ~ 3# background grays value */
}xmedia_ive_work_bg_pix_s;
```

#### 【成员】

| 成员名称          | 描述                                            |
|---------------|-----------------------------------------------|
| u8q4f4_mean   | 0 号工作背景灰度值，高 12 位为 0 号工作背景像素值，低 4 位为工作背景状态标识。 |
| u16_acc_time  | 用于工作背景有效性检查的背景累计访问时间。                         |
| u8_pre_gray   | 前一帧像素灰度值。                                     |
| u5q3_diff_thr | 差分阈值。                                         |
| u8_acc_flag   | 工作背景访问标识。                                     |
| u8_bg_gray[3] | 1 ~ 3 号背景灰度值。                                 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_candi\\_bg\\_pix\\_s](#)

[xmedia\\_ive\\_bg\\_model\\_pix\\_s](#)

[xmedia\\_ive\\_bg\\_life\\_s](#)

#### 【说明】

定义背景生命力数据。

#### 【定义】

```
typedef struct xmedia_ive_bg_life_s
{
 xmedia_u8 u8_work_bg_life[3]; /*1# ~ 3# background vitality */
 xmedia_u8 u8_candi_bg_life; /*Candidate background vitality */
}xmedia_ive_bg_life_s;
```

#### 【成员】

| 成员名称               | 描述              |
|--------------------|-----------------|
| u8_work_bg_life[3] | 1 ~ 3 号工作背景生命力。 |
| u8_candi_bg_life   | 候选背景生命力。        |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_bg\\_model\\_pix\\_s](#)

xmedia\_ive\_bg\_model\_pix\_s

#### 【说明】

定义背景模型数据。

#### 【定义】

```
typedef struct xmedia_ive_bg_model_pix_s
{
 xmedia_ive_work_bg_pix_s st_work_bg_pixel; /*Working background */
 xmedia_ive_candi_bg_pix_s st_candi_pixel; /*Candidate background */
 xmedia_ive_bg_life_s st_bg_life; /*Background vitality */
}xmedia_ive_bg_model_pix_s;
```

#### 【成员】

| 成员名称             | 描述      |
|------------------|---------|
| st_work_bg_pixel | 工作背景数据。 |
| st_candi_pixel   | 候选背景数据。 |
| st_bg_life       | 背景生命力。  |

#### 【注意事项】

无

#### 【相关数据类型及接口】

[xmedia\\_ive\\_candi\\_bg\\_pix\\_s](#)

[xmedia\\_ive\\_work\\_bg\\_pix\\_s](#)

[xmedia\\_ive\\_bg\\_life\\_s](#)

xmedia\_ive\_fg\_stat\_data\_s

#### 【说明】

定义前景状态数据。

#### 【定义】

```
typedef struct xmedia_ive_fg_stat_data_s
{
 xmedia_u32 u32_pix_num;
 xmedia_u32 u32_sum_lum;
 xmedia_u8 u8_reserved[8];
}xmedia_ive_fg_stat_data_s;
```

#### 【成员】

| 成员名称           | 描述              |
|----------------|-----------------|
| u32_pix_num    | 前景像素数目。         |
| u32_sum_lum    | 输入图像的所有像素亮度累加和。 |
| u8_reserved[8] | 保留字段。           |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

xmedia\_ive\_bg\_stat\_data\_s

#### 【说明】

定义背景状态数据。

#### 【定义】

```
typedef struct xmedia_ive_bg_stat_data_s
{
 xmedia_u32 u32_pix_num;
 xmedia_u32 u32_sum_lum;
 xmedia_u8 u8_reserved[8];
}xmedia_ive_bg_stat_data_s;
```

#### 【成员】

| 成员名称        | 描述              |
|-------------|-----------------|
| u32_pix_num | 背景像素数目。         |
| u32_sum_lum | 背景图像的所有像素亮度累加和。 |
| u8_reserved | 保留字段。           |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_match\_bg\_model\_ctrl\_s

#### 【说明】

定义背景匹配控制参数。

#### 【定义】

```
typedef struct xmedia_ive_match_bg_model_ctrl_s
{
 xmedia_u32 u32_cur_frm_num; /*Current frame timestamp, in frame units */
 xmedia_u32 u32_pre_frm_num; /*Previous frame timestamp, in frame units */
 xmedia_u16 u16_time_thr; /*Potential background replacement time threshold (range: 2 to
100 frames; default: 20) */

 xmedia_u8 u8_diff_thr_crl_coef; /*Correlation coefficients between differential threshold and
gray value (range: 0 to 5; default: 0) */
 xmedia_u8 u8_diff_max_thr; /*Maximum of background differential threshold (range: 3
to 15; default: 6) */
 xmedia_u8 u8_diff_min_thr; /*Minimum of background differential threshold (range: 3
to 15; default: 4) */
 xmedia_u8 u8_diff_thr_inc; /*Dynamic Background differential threshold increment
(range: 0 to 6; default: 0) */
 xmedia_u8 u8_fast_learn_rate; /*Quick background learning rate (range: 0 to 4; default: 2) */
 xmedia_u8 u8_det_chg_region; /*Whether to detect change region (range: 0 (no), 1 (yes);
default: 0) */
}xmedia_ive_match_bg_model_ctrl_s;
```

#### 【成员】

| 成员名称                 | 描述                                                       |
|----------------------|----------------------------------------------------------|
| u32_cur_frm_num      | 当前帧时间。                                                   |
| u32_pre_frm_num      | 前一帧时间。<br>要求 $u32\_pre\_frm\_num < u32\_cur\_frm\_num$ 。 |
| u16_time_thr         | 潜在背景替换时间阈值。<br>取值范围：[2,100]。<br>参考取值：20。                 |
| u8_diff_thr_crl_coef | 差分阈值与灰度值相关系数。<br>取值范围：[0,5]。<br>参考取值：0。                  |
| u8_diff_max_thr      | 背景差分阈值调整上限。<br>取值范围：[3,15]。<br>参考取值：6。                   |
| u8_diff_min_thr      | 背景差分阈值调整下限。<br>取值范围：[3, u8_diff_max_thr]。<br>参考取值：4。     |
| u8_diff_thr_inc      | 动态背景下差分阈值增量。<br>取值范围：[0,6]。<br>参考取值：0。                   |
| u8_fast_learn_rate   | 快速背景学习速率。<br>取值范围：[0,4]。<br>参考取值：2。                      |
| u8_det_chg_region    | 是否检测变化区域。<br>取值范围：{0,1}, 0 表示不检测, 1 表示检测。<br>参考取值：0。     |

【注意事项】

无。

## 【相关数据类型及接口】

无。

## xmedia\_ive\_update\_bg\_model\_ctrl\_s

### 【说明】

定义背景更新控制参数。

### 【定义】

```
typedef struct xmedia_ive_update_bg_model_ctrl_s
{
 xmedia_u32 u32_cur_frm_num; /*Current frame timestamp, in frame units */
 xmedia_u32 u32_pre_chk_time; /*The last time when background status is checked */
 xmedia_u32 u32_frm_chk_period; /*Background status checking period (range: 0 to 2000
frames; default: 50) */
 xmedia_u32 u32_init_min_time; /*Background initialization shortest time (range: 20 to 6000
frames; default: 100)*/
 xmedia_u32 u32_sty_bg_min_blend_time; /*Steady background integration shortest time
(range: 20 to 6000 frames; default: 200)*/
 xmedia_u32 u32_sty_bg_max_blend_time; /*Steady background integration longest time
(range: 20 to 40000 frames; default: 1500)*/
 xmedia_u32 u32_dyn_bg_min_blend_time; /*Dynamic background integration shortest time
(range: 0 to 6000 frames; default: 0)*/
 xmedia_u32 u32_static_det_min_time; /*Still detection shortest time (range: 20 to 6000 frames;
default: 80)*/
 xmedia_u16 u16_fg_max_fade_time; /*Foreground disappearing longest time (range: 1 to
255 seconds; default: 15)*/
 xmedia_u16 u16_bg_max_fade_time; /*Background disappearing longest time (range: 1 to
255 seconds; default: 60)*/
 xmedia_u8 u8_sty_bg_acc_time_rate_thr; /*Steady background access time ratio threshold
(range: 10 to 100; default: 80)*/
 xmedia_u8 u8_chg_bg_acc_time_rate_thr; /*Change background access time ratio
threshold (range: 10 to 100; default: 60)*/
 xmedia_u8 u8_dyn_bg_acc_time_thr; /*Dynamic background access time ratio
threshold (range: 0 to 50; default: 0)*/
 xmedia_u8 u8_dyn_bg_depth; /*Dynamic background depth (range: 0 to 3;
default: 3)*/
 xmedia_u8 u8_bg_eff_sta_rate_thr; /*Background state time ratio threshold when
initializing (range: 90 to 100; default: 90)*/
 xmedia_u8 u8_acce_bg_learn; /*Whether to accelerate background learning
(range: 0 (no), 1 (yes); default: 0)*/
 xmedia_u8 u8_det_chg_region; /*Whether to detect change region (range: 0 (no), 1
```

```
(yes); default: 0)*/
} xmedia_ive_update_bg_model_ctrl_s;
```

#### 【成员】

| 成员名称                      | 描述                                                                    |
|---------------------------|-----------------------------------------------------------------------|
| u32_cur_frm_num           | 当前帧时间。                                                                |
| u32_pre_chk_time          | 上次进行背景状态检查的时间点。                                                       |
| u32_frm_chk_period        | 背景状态检查时间周期。<br>取值范围：[0, 2000]。<br>参考取值：50。                            |
| u32_init_min_time         | 背景初始化最短时间。<br>取值范围：[20, 6000]。<br>参考取值：100。                           |
| u32_sty_bg_min_blend_time | 稳态背景融入最短时间。<br>取值范围：[u32_init_min_time, 6000]。<br>参考取值：200。           |
| u32_sty_bg_max_blend_time | 稳态背景融入最长时间。<br>取值范围：[u32_sty_bg_min_blend_time, 40000]。<br>参考取值：1500。 |
| u32_dyn_bg_min_blend_time | 动态背景融入最短时间。<br>取值范围：[0, 6000]。<br>参考取值：0。                             |
| u32_static_det_min_time   | 静物检测最短时间。<br>取值范围：[20, 6000]。<br>参考取值：80。                             |
| u16_fg_max_fade_time      | 前景持续消失最长时间。<br>取值范围：[1, 255]。<br>参考取值：15。                             |

| 成员名称                        | 描述                                                    |
|-----------------------------|-------------------------------------------------------|
| u16_bg_max_fade_time        | 背景持续消失最长时间。<br>取值范围：[1, 255]。<br>参考取值：60。             |
| u8_sty_bg_acc_time_rate_thr | 稳态背景访问时间比率阈值。<br>取值范围：[10, 100]。<br>参考取值：80。          |
| u8_chg_bg_acc_time_rate_thr | 变化背景访问时间比率阈值。<br>取值范围：[10, 100]。<br>参考取值：60。          |
| u8_dyn_bg_acc_time_thr      | 动态背景访问时间比率阈值。<br>取值范围：[0, 50]。<br>参考取值：0。             |
| u8_dyn_bg_depth             | 动态背景深度。<br>取值范围：[0, 3]。<br>参考取值：3。                    |
| u8_bg_eff_sta_rate_thr      | 景初始化阶段背景状态时间比率阈值。<br>取值范围：[90, 100]。<br>参考取值：90。      |
| u8_acce_bg_learn            | 是否加速背景学习。<br>取值范围：{0, 1}, 0 表示不加速, 1 表示加速。<br>参考取值：0。 |
| u8_det_chg_region           | 是否检测变化区域。<br>取值范围：{0, 1}, 0 表示不检测, 1 表示检测。<br>参考取值：0。 |

【注意事项】

要求  $u32\_init\_min\_time \leq u32\_sty\_bg\_min\_blend\_time \leq u32\_sty\_bg\_max\_blend\_time$ 。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_blob\_type\_e

#### 【说明】

定义 blob 的数据内存排布。

#### 【定义】

```
typedef enum xmedia_ive_blob_type_e
{
 XMEDIA_IVE_BLOB_TYPE_S32 = 0x0,
 XMEDIA_IVE_BLOB_TYPE_U8 = 0x1,
 /*channel = 3*/
 XMEDIA_IVE_BLOB_TYPE_YVU420SP = 0x2,
 /*channel = 3*/
 XMEDIA_IVE_BLOB_TYPE_YVU422SP = 0x3,
 XMEDIA_IVE_BLOB_TYPE_VEC_S32 = 0x4,
 XMEDIA_IVE_BLOB_TYPE_SEQ_S32 = 0x5,
 XMEDIA_IVE_BLOB_TYPE_U16 = 0x6,
 XMEDIA_IVE_BLOB_TYPE_S8 = 0x7,
 XMEDIA_IVE_BLOB_TYPE_F32 = 0x8,
 XMEDIA_IVE_BLOB_TYPE_BUTT
}xmedia_ive_blob_type_e;
```

#### 【成员】

| 成员名称                          | 描述                              |
|-------------------------------|---------------------------------|
| XMEDIA_IVE_BLOB_TYPE_S32      | Blob 数据元素为 S32 类型，参考图 1-15      |
| XMEDIA_IVE_BLOB_TYPE_U8       | Blob 数据元素为 U8 类型，参考图 1-16       |
| XMEDIA_IVE_BLOB_TYPE_YVU420SP | Blob 数据内存排布为 YVU420SP，参考图 1-17。 |
| XMEDIA_IVE_BLOB_TYPE_YVU422SP | Blob 数据内存排布为 YVU422SP，参考图 1-18。 |

| 成员名称                             | 描述                                 |
|----------------------------------|------------------------------------|
| XMEDIA_IVE_BLOB_TYPE_V<br>EC_S32 | Blob 中存储向量，每个元素为 S32 类型，参考图 1-19。  |
| XMEDIA_IVE_BLOB_TYPE_S<br>EQ_S32 | Blob 中存储序列，数据元素为 S32 类型，排布见图 1-20。 |
| XMEDIA_IVE_BLOB_TYPE_U<br>16     | Blob 数据元素为 U16 类型，参考图 1-21。        |
| XMEDIA_IVE_BLOB_TYPE_S<br>8      | Blob 数据元素为 S8 类型，参考图 1-22。         |
| XMEDIA_IVE_BLOB_TYPE_F<br>32     | Blob 数据元素为 F32 类型，参考图 1-23。        |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_blob\\_s](#)

xmedia\_ive\_blob\_s

#### 【说明】

定义多个连续存放的 blob 信息。

#### 【定义】

```
typedef struct xmedia_ive_blob_s
{
 xmedia_ive_blob_type_e en_type; /*Blob type*/
 xmedia_u32 u32stride; /*Stride, a line bytes num*/
 xmedia_u64 u64viraddr; /*virtual addr*/
 xmedia_u64 u64phyaddr; /*physical addr*/
 xmedia_u32 u32_num; /*N: frame num or sequence num,correspond to caffe
blob's n*/
 union
 {
 struct
 {
 xmedia_u32 u32width; /*W: frame width, correspond to caffe blob's w*/
 xmedia_u32 u32height; /*H: frame height, correspond to caffe blob's h*/
 };
 };
};
```

```

 xmedia_u32 u32chn; /*C: frame channel,correspond to caffe blob's c*/
 }stWhc;
 struct
 {
 xmedia_u32 u32dim; /*D: vecotr dimension*/
 xmedia_u64 u64viraddrstep; /*T: virtual adress of time steps array in each
sequence*/
 }stseq;
 }unshape;
}xmedia_ive_blob_s;

```

#### 【成员】

| 成员名称           | 描述                                          |
|----------------|---------------------------------------------|
| en_type        | Blob 类型。                                    |
| u32stride      | Blob 中单行数据的对齐后的字节数。                         |
| u64viraddr     | Blob 首虚拟地址。                                 |
| u64phyaddr     | Blob 首物理地址。                                 |
| u32num         | 表示连续内存块的数目，若一帧数据对应一个块，则表示 blob 中有 u32num 帧。 |
| u32width       | Blob 的宽度。                                   |
| u32height      | Blob 的高度。                                   |
| u32chn         | Blob 中的通道数。                                 |
| u32dim         | 向量的长度。                                      |
| u64viraddrstep | 数组地址，数组元素表示每段序列有多少个向量。                      |

#### 【注意事项】

- Caffe 中不同数据内存块用来表示内存形状的数据如下表：

| 数据块               | Dim0 | Dim1 | Dim2 | Dim3 |
|-------------------|------|------|------|------|
| Image\Feature Map | N    | C    | H    | W    |
| FC(normal) vector | N    | C    | -    | -    |
| RNN\LSTM vector   | T    | N    | D    | -    |

- 对应于本文中的 blob 表示如下表：

| 数据块               | Dim0              | Dim1     | Dim2      | Dim3     |
|-------------------|-------------------|----------|-----------|----------|
| Image\Feature Map | u32num            | 32chn    | u32height | u32width |
| FC(normal) vector | u32num            | u32width | -         | -        |
| RNN\LSTM vector   | u64viraddrstep[i] | u32num   | u32dim    | -        |

- u32Stride 表示的是在 u32Width 方向和 u32Dim 方向上一行数据对齐后的字节数。

#### 【相关数据类型及接口】

无。

### xmedia\_ive\_src\_blob\_s

#### 【说明】

定义源序列。

#### 【定义】

```
typedef xmedia_ive_blob_s xmedia_ive_src_blob_s;
```

#### 【成员】

无。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_blob\\_s](#)

[xmedia\\_ive\\_dst\\_blob\\_s](#)

### xmedia\_ive\_dst\_blob\_s

#### 【说明】

定义输出序列。

#### 【定义】

```
typedef xmedia_ive_blob_s xmedia_ive_dst_blob_s;
```

#### 【成员】

无。

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_blob\\_s](#)

[xmedia\\_ive\\_src\\_blob\\_s](#)

### xmedia\_ive\_resize\_csc\_letterbox\_op\_mode\_e

#### 【说明】

定义缩放-色彩空间转换-加边框复合任务模式枚举。

#### 【定义】

```
typedef enum xmedia_ive_resize_csc_letterbox_op_mode_e {
 XMEDIA_IVE_RCL_OP_MODE_CSC_ONLY = 0,
 XMEDIA_IVE_RCL_OP_MODE_RESIZE_LETTERBOX_ONLY = 1,
 XMEDIA_IVE_RCL_OP_MODE_RESIZE_CSC_LETTERBOX = 2,
} xmedia_ive_resize_csc_letterbox_op_mode_e;
```

#### 【成员】

| 成员名称                                         | 描述                  |
|----------------------------------------------|---------------------|
| XMEDIA_IVE_RCL_OP_MODE_CSC_ONLY              | 仅做色彩空间转换操作。         |
| XMEDIA_IVE_RCL_OP_MODE_RESIZE_LETTERBOX_ONLY | 仅做缩放-加边框操作。         |
| XMEDIA_IVE_RCL_OP_MODE_RESIZE_CSC_LETTERBOX  | 做缩放-色彩空间转换-加边框复合操作。 |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_resize\\_csc\\_letterbox\\_ctrl\\_s](#)

## xmedia\_ive\_resize\_csc\_letterbox\_csc\_mode\_e

### 【说明】

定义缩放-色彩空间转换-加边框复合任务色彩空间转换模式枚举。

### 【定义】

```
typedef enum xmedia_ive_resize_csc_letterbox_csc_mode_e {
 XMEDIA_IVE_RCL_CSC_MODE_VIDEO_BT601_YUV2RGB =
 0x0, /* CSC: YUV2RGB, video transfer mode, RGB value range [16, 235] */
 XMEDIA_IVE_RCL_CSC_MODE_VIDEO_BT709_YUV2RGB =
 0x1, /* CSC: YUV2RGB, video transfer mode, RGB value range [16, 235] */
 XMEDIA_IVE_RCL_CSC_MODE_PIC_BT601_YUV2RGB =
 0x2, /* CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255] */
 XMEDIA_IVE_RCL_CSC_MODE_PIC_BT709_YUV2RGB =
 0x3, /* CSC: YUV2RGB, picture transfer mode, RGB value range [0, 255] */
} xmedia_ive_resize_csc_letterbox_csc_mode_e;
```

### 【成员】

| 成员名称                                        | 描述                    |
|---------------------------------------------|-----------------------|
| XMEDIA_IVE_RCL_CSC_MODE_VIDEO_BT601_YUV2RGB | bt601 的 yuv2rgb 视频变换。 |
| XMEDIA_IVE_RCL_CSC_MODE_VIDEO_BT709_YUV2RGB | bt709 的 yuv2rgb 视频变换。 |
| XMEDIA_IVE_RCL_CSC_MODE_PIC_BT601_YUV2RGB   | bt601 的 yuv2rgb 图像变换。 |
| XMEDIA_IVE_RCL_CSC_MODE_PIC_BT709_YUV2RGB   | bt709 的 yuv2rgb 图像变换。 |

### 【注意事项】

无。

### 【相关数据类型及接口】

[xmedia\\_ive\\_resize\\_csc\\_letterbox\\_ctrl\\_s](#)

## xmedia\_ive\_resize\_csc\_letterbox\_ctrl\_s

### 【说明】

定义缩放-色彩空间转换-加边框复合任务控制参数。

### 【定义】

```
typedef struct xmedia_ive_resize_csc_letterbox_ctrl_s {
 xmedia_ive_resize_csc_letterbox_op_mode_e op_mode;
 xmedia_ive_resize_csc_letterbox_csc_mode_e csc_mode;
 xmedia_u16 u16_top_padding_len;
 xmedia_u16 u16_bottom_padding_len;
 xmedia_u16 u16_left_padding_len;
 xmedia_u16 u16_right_padding_len;
 xmedia_u8 u8_letterbox_value;
} xmedia_ive_resize_csc_letterbox_ctrl_s;
```

### 【成员】

| 成员名称               | 描述                       |
|--------------------|--------------------------|
| op_mode            | 缩放-色彩空间转换-加边框任务模式。       |
| csc_mode           | 缩放-色彩空间转换-加边框任务色彩空间转换模式。 |
| top_padding_len    | 输出图像上边框长度。               |
| bottom_padding_len | 输出图像下边框长度。               |
| left_padding_len   | 输出图像左边框长度。               |
| right_padding_len  | 输出图像右边框长度。               |
| letterbox_value    | 输出图像边框填充像素值。             |

### 【注意事项】

无。

### 【相关数据类型及接口】

[xmedia\\_ive\\_resize\\_csc\\_letterbox\\_op\\_mode\\_e](#)

[xmedia\\_ive\\_resize\\_csc\\_letterbox\\_csc\\_mode\\_e](#)

[xmedia\\_ive\\_acc\\_mode\\_e](#)

### 【说明】

定义 acc mask 使用模式。

### 【定义】

```
typedef enum xmedia_ive_acc_mode_e {
 XMEDIA_IVE_ACC_MODE_NO_USE_MASK = 0x0,
 XMEDIA_IVE_ACC_MODE_USE_MASK = 0x1,

 XMEDIA_IVE_ACC_MODE_BUTT
} xmedia_ive_acc_mode_e;
```

#### 【成员】

| 成员名称                            | 描述        |
|---------------------------------|-----------|
| XMEDIA_IVE_ACC_MODE_NO_USE_MASK | 不适用 mask。 |
| XMEDIA_IVE_ACC_MODE_USE_MASK    | 使用 mask。  |

#### 【注意事项】

无。

#### 【相关数据类型及接口】

[xmedia\\_ive\\_acc\\_ctrl\\_s](#)

xmedia\_ive\_acc\_ctrl\_s

#### 【说明】

定义像素累加控制数据。

#### 【定义】

```
typedef struct xmedia_ive_acc_ctrl_s {
 xmedia_ive_acc_mode_e mode;
 xmedia_u16 mask[4];
} xmedia_ive_acc_ctrl_s;
```

#### 【成员】

| 成员名称    | 描述      |
|---------|---------|
| mode    | 像素累加模式。 |
| mask[4] | Mask。   |

#### 【注意事项】

无。

【相关数据类型及接口】

[xmedia\\_ive\\_acc\\_mode\\_e](#)

仅限深圳市安远威视科技有限公司使用

# 4 错误码

## 4.1 IVE 错误码

智能加速引擎 API 错误码如表 4-1 所示。

表4-1 智能加速引擎 API 错误码

| 错误代码       | 宏定义                              | 描述         |
|------------|----------------------------------|------------|
| 0x80010001 | XMEDIA_ERRCODE_INVALID_PARAM     | 参数超出合法范围   |
| 0x80010002 | XMEDIA_ERRCODE_NULL_PTR          | 函数参数中存在空指针 |
| 0x80010003 | XMEDIA_ERRCODE_NOT_ENOUGH_MEMORY | 内存不足       |
| 0x80010004 | XMEDIA_ERRCODE_NOT_INIT          | 模块未初始化     |
| 0x80010005 | XMEDIA_ERRCODE_NOT_SUPPORT       | 不支持的参数或者功能 |
| 0x80010006 | XMEDIA_ERRCODE_TIMEOUT           | 超时         |
| 0x80010007 | XMEDIA_ERRCODE_BAD_ADDRESS       | 不可用的地址     |
| 0x80010008 | XMEDIA_ERRCODE_OPEN_FAILED       | 打开文件失败     |
| 0x80010009 | XMEDIA_ERRCODE_CLOSE_FAILED      | 关闭文件失败     |
| 0x8001000A | XMEDIA_ERRCODE_CREATE_FAILED     | 创建文件失败     |
| 0x8001000B | XMEDIA_ERRCODE_DESTROY_FAILED    | 销毁文件失败     |

| 错误代码       | 宏定义                            | 描述                     |
|------------|--------------------------------|------------------------|
| 0x8001000C | XMEDIA_ERRCODE_BUFFER_EMPTY    | 节点资源用尽                 |
| 0x8001000D | XMEDIA_ERRCODE_BUFFER_FULL     | 缓冲区中数据满                |
| 0x8001000E | XMEDIA_ERRCODE_NO_BUFFER_FREE  | 分配缓存失败，如申请的数据缓冲区太大     |
| 0x8001000F | XMEDIA_ERRCODE_NOT_PERMITTED   | 操作不允许，状态机不对            |
| 0x80010010 | XMEDIA_ERRCODE_EXIST           | 试图申请或者创建已经存在的设备、通道或者资源 |
| 0x80010011 | XMEDIA_ERRCODE_NOT_EXIST       | 试图使用或者销毁不存在的设备、通道或者资源  |
| 0x80010012 | XMEDIA_ERRCODE_NOT_READY       | 驱动未加载                  |
| 0x80010013 | XMEDIA_ERRCODE_BUSY            | 系统忙                    |
| 0x80010014 | XMEDIA_ERRCODE_COPY_DATA_ERROR | 拷贝数据失败                 |
| 0x80010015 | XMEDIA_ERRCODE_TYPE_ERROR      | 数据类型有误                 |
| 0x80010016 | XMEDIA_ERRCODE_CHN_FULL        | 通道满                    |
| 0x80010017 | XMEDIA_ERRCODE_INVALID_CHN_ID  | 通道组号错误或无效区域句柄          |
| 0x80010018 | XMEDIA_ERRCODE_INVALID_DEV_ID  | 设备号错误或无效区域句柄           |

| 错误代码       | 宏定义                            | 描述           |
|------------|--------------------------------|--------------|
| 0x80010019 | XMEDIA_ERRCODE_INVALID_PIPE_ID | 管道号错误或无效区域句柄 |
| 0x8001001A | XMEDIA_ERRCODE_INVALID_GRP_ID  | Grp 号无效      |
| 0x8001001B | XMEDIA_ERRCODE_NOT_CONFIG      | 使用前未配置       |
| 0x8001001C | XMEDIA_ERRCODE_NOT_ENABLE      | 未使能          |
| 0x8001001D | XMEDIA_ERRCODE_NOT_DISABLE     | 未禁用          |
| 0x8001001E | XMEDIA_ERRCODE_NOT_BIND        | 未绑定          |
| 0x8001001F | XMEDIA_ERRCODE_BINDED          | 已绑定          |

# 5 Proc 调试信息

## 5.1 概述

调试信息采用了 Linux 下的 proc 文件系统，可实时反映当前系统的运行状态，所记录的信息可供问题定位及分析时使用。

### 【文件目录】

/proc/umap

### 【信息查看方法】

- 在控制台上可以使用 cat 命令查看信息，cat /proc/umap/ive；也可以使用其他常用的文件操作命令，例如 cp /proc/umap/ive ./，将文件拷贝到当前目录。
- 在应用程序中可以将上述文件当作普通只读文件进行读操作，例如 fopen、fread 等。



说明

参数在描述时有以下 2 种情况需要注意：

- 取值为{0, 1}的参数，如未列出具体取值和含义的对应关系，则参数为 1 时表示肯定，为 0 时表示否定。
- 取值为{aaa, bbb, ccc}的参数，未列出具体取值和含义的对应关系，但可直接根据取值 aaa、bbb 或 ccc 判断参数含义。

## 5.2 Proc 信息说明

### 【调试信息】

| -----module param----- |              |
|------------------------|--------------|
| save_power             | max_node_num |
| 0                      | 20           |

```

-----ive queue info-----
 wait busy wait_cur_id wait_end_id busy_cur_id busy_end_id
 1 -1 0 0 0 0

-----ive task info-----
 hnd task_finish last_id task_id handle_wrap finish_wrap
 0 0 0 0 0 0

-----ive run-time info-----
last_inst cnt_per_sec max_cnt_per_sec total_int_cnt_last_sec total_int_cnt qt_cnt st_cnt
 1 1 2 4 5 0 0

cost_tm max_cost_tm cost_tm_per_sec max_cost_tm_per_sec total_int_cost_tm run_tm
 7 17 17 23 64 20

-----ive invoke info-----
dma filter csc flt_csc sobel mag_ang dilate erode
 5 0 0 0 0 0 0

thresh and sub or integ hist thresh_s16 thresh_u16
 0 0 0 0 0 0 0

16to8 ord_statflt bern_sen map equal_hist add xor ncc
 0 0 0 0 0 0 0

ccl gmm canny lbp norm_grad lk shi_to_masi grad_fg
 0 0 0 0 0 0 0

match_mod update_mod radon ann svm adp_thr lineflt_h noise_rm_h
 0 0 0 0 0 0 0

plate_char sad gmm2 resize cnn persp_trans kcf hog
 0 0 0 0 0 0 0

xnn resize_csc_lb
 0 0

```

#### 【调试信息分析】

记录当前 ive 工作状态资源信息和算子调用信息。

#### 【参数说明】

| 参数                               |                        | 描述                                     |
|----------------------------------|------------------------|----------------------------------------|
| module param<br>模块参数             | save_power             | 低功耗标志。<br>0：关闭低功耗；<br>1：打开低功耗。         |
|                                  | max_node_num           | 队列最大节点数。<br>取值范围[8, 512]。              |
| ive queue info<br>ive 队列信息       | wait                   | 等待队列编号（0 或 1）。                         |
|                                  | busy                   | 正在调度队列编号（0, 1 或-1），<br>-1 表示 IVE 硬件空闲。 |
|                                  | wait_cur_id            | 等待队列的首个有效任务 id。                        |
|                                  | wait_end_id            | 等待队列的末尾有效任务 id + 1。                    |
|                                  | busy_cur_id            | 调度队列的首个有效任务 id。                        |
|                                  | busy_end_id            | 调度队列的末尾有效任务 id + 1。                    |
| ive task info<br>ive task 相关信息   | hnd                    | 当前可分配的任务 handle 号。                     |
|                                  | task_finish            | 当前已完成任务的个数。                            |
|                                  | last_id                | 上一次中断完成的任务 id。                         |
|                                  | task_id                | 本次中断完成的任务 id。                          |
|                                  | handle_wrap            | 用户 handle 号分配发生回写的次数。                  |
|                                  | finish_wrap            | 完成任务数发生回写的次数。                          |
| ive run-time info<br>ive 运行时相关信息 | last_inst              | 用户最后一次提交任务时传入的 b_instant 值。            |
|                                  | cnt_per_sec            | 最近一次的 1 秒内中断执行次数。                      |
|                                  | max_cnt_per_sec        | 历史上的 1 秒内最大的中断执行次数。                    |
|                                  | total_int_cnt_last_sec | 上一秒上报中断总次数。                            |
|                                  | total_int_cnt          | ive 产生中断的总次数。                          |

| 参数                          |                     | 描述                         |
|-----------------------------|---------------------|----------------------------|
|                             | qt_cnt              | 查询 ive 链表超时次数。             |
|                             | st_cnt              | ive 系统超时次数。                |
|                             | cost_tm             | 最近一次执行中断的执行耗时。<br>单位：us    |
|                             | max_cost_tm         | 执行一次中断的最大耗时。<br>单位：us      |
|                             | cost_tm_per_pec     | 最近一秒执行中断的执行耗时。<br>单位：us    |
|                             | max_cost_tm_per_sec | 历史上一秒执行中断的最大执行耗时。<br>单位：us |
|                             | total_int_cost_tm   | 中断处理总时间。<br>单位：us          |
|                             | run_tm              | ive 运行总时间。<br>单位：s         |
| ive invoke info<br>ive 调用信息 | dma                 | dma 的调用次数。其它不再一一列出。        |

【注意】

建议代码调试阶段关闭低功耗，调试完成再打开低功耗。