



AX IVES API 文档

文档版本：V1.2

发布日期：2024/04/30

目 录

前 言	5
修订历史.....	6
1 概述.....	7
1.1 概述.....	7
1.2 功能描述.....	7
1.2.1 MD	7
1.2.2 OD.....	8
1.2.3 SCD	9
1.2.4 API 调用流程.....	10
2 API 参考	11
2.1 初始化	11
2.1.1 AX_IVES_Init.....	11
2.1.2 AX_IVES_Delinit	13
2.2 MD	14
2.2.1 AX_IVES_MD_Init.....	14
2.2.2 AX_IVES_MD_Delinit	15
2.2.3 AX_IVES_MD_CreateChn	16
2.2.4 AX_IVES_MD_DestoryChn	17
2.2.5 AX_IVES_MD_SetChnAttr	18
2.2.6 AX_IVES_MD_GetChnAttr	19
2.2.7 AX_IVES_MD_Process	20
2.3 OD	21
2.3.1 AX_IVES_OD_Init	21

2.3.2 AX_IVES_OD_Delnit.....	23
2.3.3 AX_IVES_OD_CreateChn.....	24
2.3.4 AX_IVES_OD_DestoryChn	25
2.3.5 AX_IVES_OD_SetChnAttr.....	26
2.3.6 AX_IVES_OD_GetChnAttr	27
2.3.7 AX_IVES_OD_Process	28
2.4 SCD.....	30
2.4.1 AX_IVES_SCD_Init	30
2.4.2 AX_IVES_SCD_Delnit.....	31
2.4.3 AX_IVES_SCD_CreateChn	32
2.4.4 AX_IVES_SCD_DestoryChn.....	33
2.4.5 AX_IVES_SCD_SetChnAttr	34
2.4.6 AX_IVES_SCD_GetChnAttr	35
2.4.7 AX_IVES_SCD_Process	36
3 数据结构	38
AXIVES_MAX_IMAGE_WIDTH.....	38
AXIVES_MAX_IMAGE_HEIGHT	38
AXIVES_RECT_T.....	39
AXIVES_MB_SIZE_T	40
AXIVES_IMAGE_T.....	40
MD_CHN.....	41
AX_MD_ALG_MODE_E.....	41
AX_MD_CHN_ATTR_T	42
AX_MD_MB_THR_T.....	43
AX_MD_MB_SAD_T.....	44
OD_CHN	45

AX_OD_CHN_ATTR_T	45
AX_IVES_OD_IMAGE_T	46
AX_IVES_CCL_MODE_E.....	47
AX_IVES_REGION_T	50
AX_IVES_CCBLOB_T.....	51
SCD_CHN.....	52
AX_SCD_CHN_ATTR_T.....	52
AX_IVES_BLK_SIZE_T.....	53
4 错误码.....	55
5 日志	56

权利声明

爱芯元智半导体股份有限公司或其许可人保留一切权利。

非经权利人书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

注意

您购买的产品、服务或特性等应受商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非商业合同另有约定，本公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

前 言

本文档主要介绍 AX IVES 的 API 接口说明。

适用产品

AX620E 系列产品（AX630C、AX620Q）

适读人群

- 终端用户
- 售前
- 售后
- 技术人员

符号与格式定义

符号/格式	说明
xxx	表示您可以执行的命令行。
斜体	表示变量。如，“安装目录/AX620E_SDK_Vx.x.x/build 目录”中的“安装目录”是一个变量，由您的实际环境决定。
☞ 说明/备注：	表示您在使用产品的过程中，我们向您说明的事项。
！ 注意：	表示您在使用产品的过程中，需要您特别注意的事项。

修订历史

文档版本	发布时间	修订说明
V1.0	2023/08/15	Initial
V1.0	2024/01/25	更新平台说明
V1.1	2024/01/30	AX_IVES_MD_SetChnAttr 增加参数说明
V1.2	2024/04/30	MD 增加残差绝对值 SAD 的支持

1 概述

本章节将对 AX IVES 模块做简要介绍，包含模块涉及的重要概念。

1.1 概述

IVES (Intelligent Video Extension System)是本公司提供的更高层次的智能视频监控应用 API，用户基于 IVES 可以快速开发出相关智能应用，当前 IVES 支持的智能应用有：MD (Motion Detection 移动侦测)，OD (Occlusion Detection 遮挡侦测)、SCD(Scene Change Detection 场景切换侦测)。

1.2 功能描述

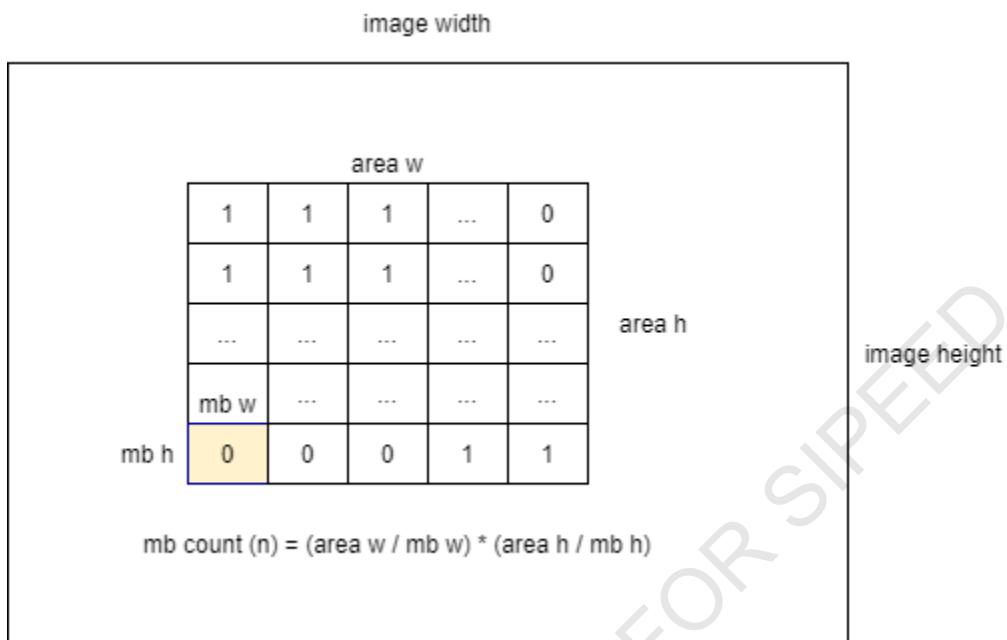
1.2.1 MD

移动侦测通过检测视频的亮度变化，侦测视频的运动状态，得出分析结果。

在 MD 中，一个区域 ([AXIVES_RECT_T](#)) 对应一个通道([MD_CHN](#))。

基本原理

- MD 将待检测区域划分成一个个的宏块，分别计算各个宏块的 Y 的平均值(mean)。
- 依次比较前后帧各个宏的 Y 平均值的差值是否大于 Y 阈值([u8ThrY](#))，得出宏块二值化结果，其中 $\text{abs}(Y_{\text{diff}}) > u8ThrY = 1$: 标识移动， 0: 标识未移动
- 调用者通过判断宏块二值化结果来决策判定视频帧是否有移动。



1.2.2 OD

遮挡侦测通过检测视频的亮度变化来判断是否有遮挡。

在 OD 中，一个区域 ([AXIVES_RECT_T](#)) 对应一个通道([OD_CHN](#))。

基本原理

- 首先统计区域内 Y 低于阈值亮度([u8ThrdY](#))的个数，若个数超过区域内像素总个数 * [u8ConfidenceY](#) / 100.0，则判断该区域被遮挡。
- 若当前视频帧的 LUX 低于阈值亮度([u32LuxThrd](#))，则判断该区域被遮挡。
- 若前后视频帧的 LUX 超过阈值([u32LuxDiff](#))，则判断该区域被遮挡。

！注意：

LUX 精度为 U22. 10，即 1024 标识 1 LUX

1.2.3 SCD

场景切换侦测通过计算前后帧图像的相关性来判断是否有场景切换。

在 SCD 中，一个区域 ([AXIVES_RECT_T](#)) 对应一个通道([SCD_CHN](#))。

基本原理

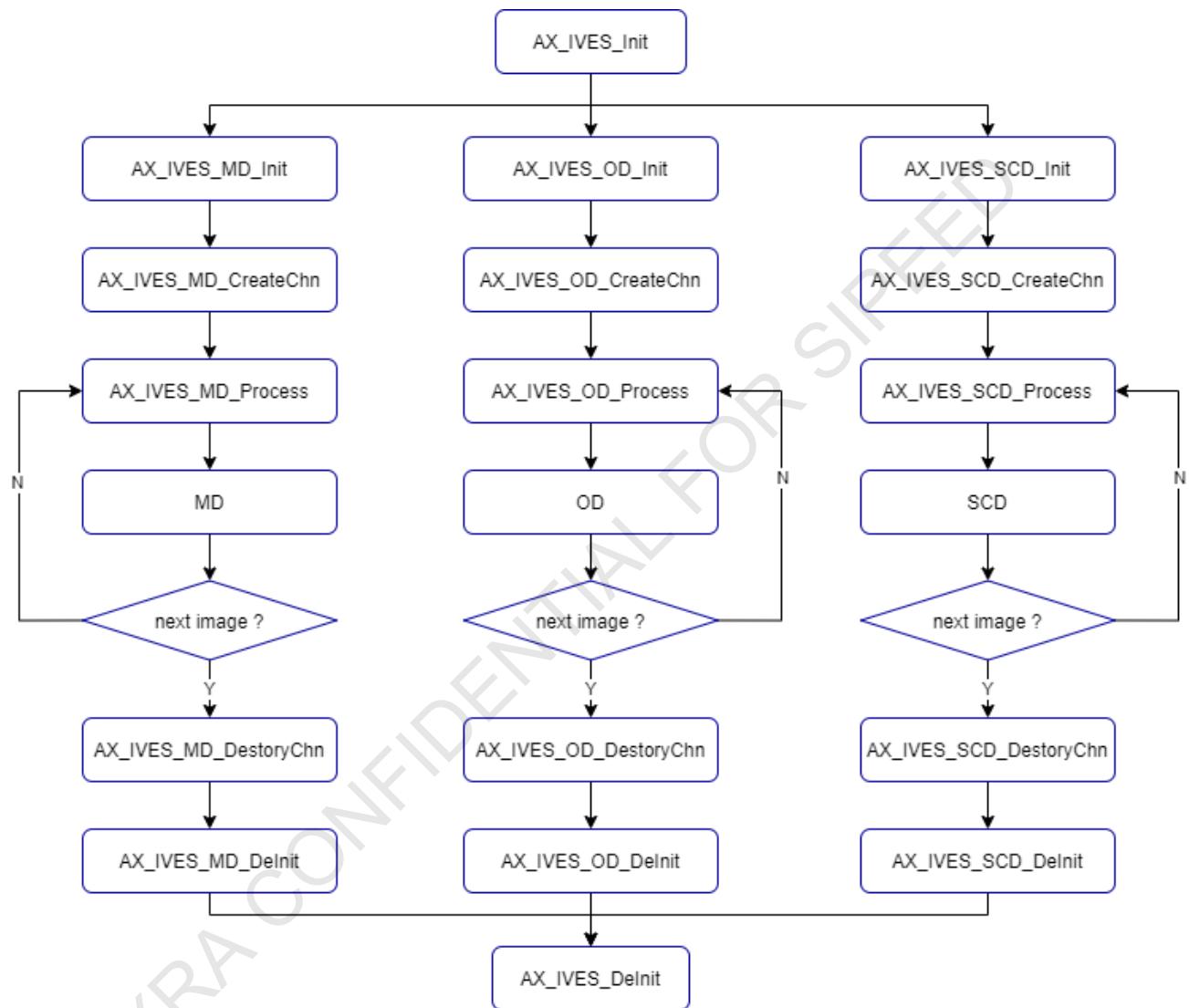
- 默认将区域划分成 4x3 个 [BLOCK](#)，总共 12 个 blk，记为 blkCnt;
- 依次对每个 BLOCK 统计灰度直方图 hist1, hist2, ..., hist12;
- 依次对前后帧图像的直方图计算距离，得出相关性结果 d1,d2, ..., d12;
- 依次将 d1, d2, ..., d12 和 [u8Thrd](#) 阈值比较，统计小于 u8Thrd 的个数 cnt;
- 若 cntblkCnt*100 大于 [u8Confidence](#)，则标记场景未切换(0)，否则标记切换(1)。

！注意：

- BLOCK 的最小像素为 64x64，若不满足，则区域不划分 BLOCK，即 blkCnt=1
- 支持通过环境变量修改 BLOCK 划分

```
export IVES_SCD_BLK_ROW=4  
export IVES_SCD_BLK_COL=3
```

1.2.4 API 调用流程



2 API 参考

2.1 初始化

2.1.1 AXIVES_Init

【描述】

对 IVES 模块初始化，日志配置。

【语法】

```
AX_S32 AXIVES_Init(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件：ax_ives_api.h
- 库文件：libax_ives.so

【注意】

- 本接口支持重复调用。
- 本接口和 AXIVES_DeInit 一一对应，不支持线程重入，调用者在需要进行多线程的调用流程管理

【举例】

无

【相关主题】

函数: [AXIVES_DeInit](#)

2.1.2 AXIVES_DeInit

【描述】

对 IVES 模块去初始化。

【语法】

```
AX_S32 AXIVES_DeInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_Init](#)

2.2 MD

2.2.1 AXIVESMDInit

【描述】

对 MD 模块初始化。

【语法】

```
AX_S32 AXIVESMDInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVESMDDeInit](#)

2.2.2 AXIVESMDDeInit

【描述】

对 MD 模块去初始化。

【语法】

```
AX_S32 AXIVESMDDeInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVESMDInit](#)

2.2.3 AXIVESMDCreateChn

【描述】

创建 MD 通道

【语法】

```
AX_S32 AXIVESMDCreateChn(MD CHN mdChn, AX MD CHN ATTR T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
mdChn	MD 通道, 大于或等于 0	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败, 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVESMDDestoryChn](#)

2.2.4 AXIVESMDDestoryChn

【描述】

销毁 MD 通道

【语法】

AX_S32 AXIVESMDDestoryChn ([MD CHN](#) mdChn)

【参数】

参数名称	描述	输入/输出
mdChn	MD 通道	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVESMDCreateChn](#)

2.2.5 AXIVESMD_SetChnAttr

【描述】

更新 MD 通道属性

【语法】

```
AX_S32 AXIVESMD_SetChnAttr(MD_CHN mdChn, AX MD CHN ATTR T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
mdChn	MD 通道	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

参数 pstAttr:

- 不能动态设置的参数: enAlgMode、stMbSize、stArea

【举例】

无

【相关主题】

函数: [AXIVESMD_GetChnAttr](#)

2.2.6 AXIVESMDGetChnAttr

【描述】

获取 MD 通道属性

【语法】

```
AX_S32 AXIVESMDGetChnAttr(MD CHN mdChn, AX MD CHN ATTR T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
mdChn	MD 通道	输入
pstAttr	通道属性	输出

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVESMD_SetChnAttr](#)

2.2.7 AXIVESMD_Process

【描述】

图像 MD 处理。

【语法】

```
AX_S32 AXIVESMD_Process(MD_CHN mdChn, AXIVES_IMAGE_T *pstCur,  
AX_MD_MB_THR_T *pstMbThr, AX_MD_MB_SAD_T *pstMbSad, AXIVES_CCBLOB_T  
*pstBlob)
```

【参数】

参数名称	描述	输入/输出
mdChn	MD 通道	输入
pstCur	当前图像帧指针	输入
pstMbThr	返回处理结果	输出
pstMbSad	返回残差绝对值结果	输出
pstBlob	返回连通区域信息	输出

【返回值】

返回值	描述
0	成功
非 0	失败, 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

2.3 OD

2.3.1 AXIVESODInit

【描述】

对 OD 模块初始化。

【语法】

```
AX_S32 AXIVESODInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件：ax_ives_api.h
- 库文件：libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_DeInit](#)

2.3.2 AXIVES_OD_DeInit

【描述】

对 OD 模块去初始化。

【语法】

```
AX_S32 AXIVES_OD_DeInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_Init](#)

2.3.3 AXIVES_OD_CreateChn

【描述】

创建 OD 通道

【语法】

```
AX_S32 AXIVES_OD_CreateChn(OD CHN odChn, AX OD CHN ATTR T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
odChn	OD 通道, 大于或等于 0	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败, 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_DestoryChn](#)

2.3.4 AXIVES_OD_DestroyChn

【描述】

销毁 OD 通道

【语法】1

AX_S32 AXIVES_OD_DestroyChn ([OD_CHN](#) odChn)

【参数】

参数名称	描述	输入/输出
odChn	OD 通道	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_CreateChn](#)

2.3.5 AXIVES_OD_SetChnAttr

【描述】

更新 OD 通道属性

【语法】

```
AX_S32 AXIVES_OD_SetChnAttr(OD_CHN odChn, AX_OD_CHN_ATTR_T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
odChn	OD 通道	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_GetChnAttr](#)

2.3.6 AXIVES_OD_GetChnAttr

【描述】

获取 OD 通道属性

【语法】

```
AX_S32 AXIVES_OD_GetChnAttr(OD_CHN odChn, AX_OD_CHN_ATTR_T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
odChn	OD 通道	输入
pstAttr	OD 通属性	输出

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_OD_SetChnAttr](#)

2.3.7 AXIVES_OD_Process

【描述】

图像 OD 处理

【语法】

```
AX_S32 AXIVES_OD_Process(OD_CHN odChn, const AXIVES_OD_IMAGE_T *pstCur,  
AX_U8 *pResult)
```

【参数】

参数名称	描述	输入/输出
odChn	OD 通道	输入
pstCur	当前图像帧指针	输入
pResult	返回处理结果， 1: 遮挡， 0: 不遮挡	输出

【返回值】

返回值	描述
0	成功
非 0	失败， 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

无

AEXRA CONFIDENTIAL FOR SPEED

2.4 SCD

2.4.1 AXIVES_SCD_Init

【描述】

对 SCD 始化。

【语法】

```
AX_S32 AXIVES_SCD_Init(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_DeInit](#)

2.4.2 AXIVES_SCD_DeInit

【描述】

对 SCD 模块去初始化。

【语法】

```
AX_S32 AXIVES_SCD_DeInit(AX_VOID)
```

【参数】

无

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_Init](#)

2.4.3 AXIVES_SCD_CreateChn

【描述】

创建 SCD 通道

【语法】

```
AX_S32 AXIVES_SCD_CreateChn(SCD CHN scdChn, AX SCD CHN ATTR T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
scdChn	SCD 通道, 大于或等于 0	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败, 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_DestoryChn](#)

2.4.4 AXIVES_SCD_DestoryChn

【描述】

销毁 SCD 通道

【语法】1

AX_S32 AXIVES_SCD_DestoryChn([SCD_CHN](#) scdChn)

【参数】

参数名称	描述	输入/输出
scdChn	SCD 通道	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_CreateChn](#)

2.4.5 AXIVES_SCD_SetChnAttr

【描述】

更新 SCD 通道属性

【语法】

```
AX_S32 AXIVES_SCD_SetChnAttr(SCD_CHN odChn, AX_SCD_CHN_ATTR_T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
scdChn	SCD 通道	输入
pstAttr	通道属性	输入

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_GetChnAttr](#)

2.4.6 AXIVES_SCD_GetChnAttr

【描述】

获取 SCD 通道属性

【语法】

```
AX_S32 AXIVES_SCD_GetChnAttr(SCD_CHN scdChn, AX_SCD_CHN_ATTR_T *pstAttr)
```

【参数】

参数名称	描述	输入/输出
scdChn	SCD 通道	输入
pstAttr	SCD 通属性	输出

【返回值】

返回值	描述
0	成功
非 0	失败，返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

函数: [AXIVES_SCD_SetChnAttr](#)

2.4.7 AXIVES_SCD_Process

【描述】

图像 SCD 处理

【语法】

```
AX_S32 AXIVES_SCD_Process(SCD_CHN scdChn, const AXIVES_IMAGE_T *pstImg,  
AX_U8 *pResult)
```

【参数】

参数名称	描述	输入/输出
scdChn	SCD 通道	输入
pstCur	当前图像帧指针	输入
pResult	返回处理结果, 1: 切换, 0: 未切换	输出

【返回值】

返回值	描述
0	成功
非 0	失败, 返回错误码

【需求】

- 头文件: ax_ives_api.h
- 库文件: libax_ives.so

【注意】

无

【举例】

无

【相关主题】

无

AEXRA CONFIDENTIAL FOR SPEED

3 数据结构

AXIVESMAXIMAGEWIDTH

【说明】

IVES 模块支持的最大图像宽度

【定义】

```
#define AXIVESMAXIMAGEWIDTH (4096)
```

【成员】

无

【注意】

无

【相关数据类型及接口】

无

AXIVESMAXIMAGEHEIGHT

【说明】

IVES 模块支持的最大图像高度

【定义】

```
#define AXIVESMAXIMAGEHEIGHT (3072)
```

【成员】

无

【注意】

无

【相关数据类型及接口】

无

AXIVES_RECT_T**【说明】**

区域定义

【定义】

```
typedef struct axIVES_RECT_T {  
    AX_U32 u32X;  
    AX_U32 u32Y;  
    AX_U32 u32W;  
    AX_U32 u32H;  
} AXIVES_RECT_T;
```

【成员】

成员名称	描述
u32X	区域起始横坐标
u32Y	区域起始纵坐标
u32W	区域宽度
u32H	区域高度

【注意】

无

【相关数据类型及接口】

无

AXIVES_MB_SIZE_T

【说明】

宏块大小定义

【定义】

```
typedef struct axIVES_MB_SIZE_T {  
    AX_U32 u32W; /* x pixels */  
    AX_U32 u32H; /* y pixels */  
} AXIVES_MB_SIZE_T;
```

【成员】

成员名称	描述
u32W	宏块宽度
u32H	宏块高度

【注意】

无

【相关数据类型及接口】

无

AXIVES_IMAGE_T

【说明】

IVES 图像帧定义

【定义】

```
typedef AX_VIDEO_FRAME_T AXIVES_IMAGE_T
```

【成员】

参见多媒体 AX_VIDEO_FRAME_T 结构体定义

【注意】

无

【相关数据类型及接口】

无

MD_CHN

【说明】

MD 通道定义

【定义】

```
typedef AX_S32 MD_CHN
```

【成员】

无

【注意】

无

【相关数据类型及接口】

无

AX_MD_ALG_MODE_E

【说明】

MD 模式定义

【定义】

```
typedef enum axMD_ALG_MODE_E {  
  
    AX_MD_MODE_REF = 0x0,  
  
    AX_MD_MODE_BUTT  
  
} AX_MD_ALG_MODE_E;
```

【成员】

成员名称	描述
AX_MD_MODE_REF	以上一帧作为计算参考帧

【注意】

无

【相关数据类型及接口】

无

AX_MD_CHN_ATTR_T

【说明】

MD 通道属性

【定义】

```
typedef struct axMD_CHN_ATTR_T {  
  
    MD_CHN mdChn;  
  
    AX_MD_ALG_MODE_E enAlgMode;  
  
    AXIVES_MB_SIZE_T stMbSize;
```

```

AXIVES\_RECT\_T stArea;
AX_U8 u8ThrY; /* threshold of Y */

} AX_MD_CHN_ATTR_T;

```

【成员】

成员名称	描述
mdChn	MD 通道，非负整数。
enAlgMode	模式，目前仅支持 AX_MODE_REF
stMbSize	宏块大小
stArea	区域
u8ThrY	Y 阈值

【注意】

- 区域位置和宽高要落在图像帧之内，不能超过图像帧范围。
- 区域宽高要能分别被宏块宽高整除。

【相关数据类型及接口】

无

AX_MD_MB_THR_T

【说明】

MD 二值化结果定义

【定义】

```

typedef struct axMD_MB_THR_T {
    AX_U32 u32Count; /* MB count */
    AX_U8 *pMbThrs; /* 1 or 0 threshold of each MB */
} AX_MD_MB_THR_T;

```

【成员】

成员名称	描述
u32Count	宏块个数
pMbThrs	每个宏块的 MD 二值化结果 1: 移动 0: 不移动

【注意】

无

【相关数据类型及接口】

无

AX_MD_MB_SAD_T

【说明】

MD 残差绝对值 SAD(Sum of Difference)结果定义

【定义】

```
typedef struct axMD_MB_SAD_T {  
    AX_U32 u32Count; /* MB count */  
    AX_U8 *pMbSad; /* 0 ~ 255 of each MB */  
} AX_MD_MB_SAD_T;
```

【成员】

成员名称	描述
u32Count	宏块个数
pMbSad	每个宏块的 MD 残差绝对值结果

【注意】

无

【相关数据类型及接口】

无

OD_CHN

【说明】

OD 通道定义

【定义】

```
typedef AX_S32 OD_CHN
```

【成员】

无

【注意】

无

【相关数据类型及接口】

无

AX_OD_CHN_ATTR_T

【说明】

OD 通道属性定义

【定义】

```
typedef struct axOD_CHN_ATTR_T {  
    OD_CHN odChn;  
    AXIVES_RECT_T stArea; /* area of OD */
```

```

AX_U32 u32FrameRate; /* frame rate: 25/30/60 */

AX_U8 u8ThrdY; /* threshold of Y, range [0 - 255] */

AX_U8 u8ConfidenceY; /* threshold confidence percent, range [0 - 100] */

AX_U32 u32LuxThrd; /* lux of AE, U22.10 that mean 1 lux = 1024 */

AX_U32 u32LuxDiff; /* lux diff threshold, U22.10 */

} AX_OD_CHN_ATTR_T;

```

【成员】

成员名称	描述
odChn	OD 通道
stArea	区域
u32FrameRate	帧率
u8ThrdY	Y 阈值
u8ConfidenceY	Y 阈值比例门限, 范围[0 - 100], 表示区域内 Y 小于 u8ThrdY 阈值的个数相对于区域像素总 个数的百分比(0 - 100%)
u32LuxThrd	LUX 阈值, 精度 22.10, 即 1024 表示 1 LUX
u32LuxDiff	LUX 差值, 精度 22.10, 即 1024 表示 1 LUX

【注意】

无

【相关数据类型及接口】

无

AXIVES_OD_IMAGE_T**【说明】**

OD 图像帧定义

【定义】

```
typedef struct axIVES_OD_IMAGE_T {  
    AXIVES_IMAGE_T *pstImg;  
    AX_U32 u32Lux; /* current lux returned by AE, U22.10 */  
} AXIVES_OD_IMAGE_T;
```

【成员】

成员名称	描述
pstImg	图像帧结构定义
u32Lux	当前图像帧的 LUX 亮度，精度 U22.10，即 1024 表示 1 LUX

【注意】

无

【相关数据类型及接口】

无

AXIVES_CCL_MODE_E

【说明】

定义连通区域模式

【定义】

```
typedef enum {  
    IVES_CCL_MODE_4C = 0x0, /* 4-connectivity */
```

```
IVES_CCL_MODE_8C = 0x1, /* 8-connectivity */  
  
IVES_CCL_MODE_BUTT  
  
} AX_IVES_CCL_MODE_E;
```

【成员】

成员名称	描述
IVES_CCL_MODE_4C	4 – 连通
IVES_CCL_MODE_8C	8 – 连通

【注意】

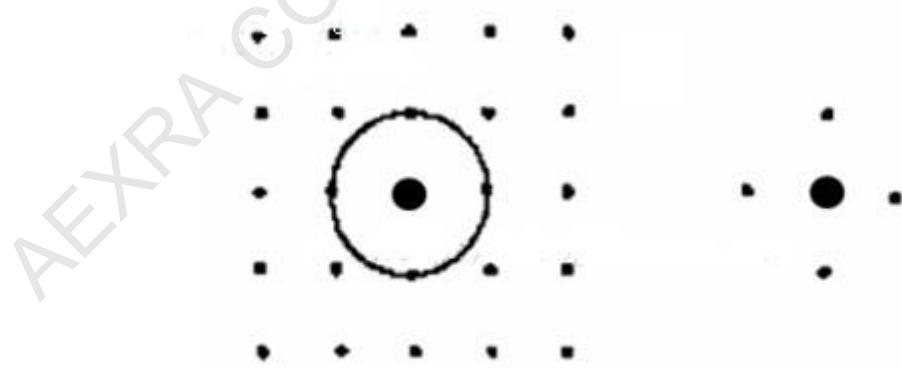
无

【相关数据类型及接口】

无

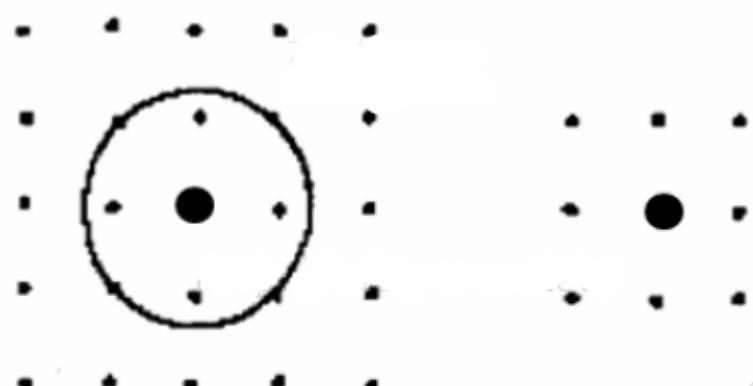
【连通区域】

所谓四连通区域或四邻域，是指对应像素位置的上、下、左、右，是紧邻的位置。共 4 个方向，所以称之为四连通区域，又叫四邻域。



4 连通区域

所谓八连通区域或八邻域，是指对应位置的上、下、左、右、左上、右上、左下、右下，是紧邻的位置和斜向相邻的位置。共 8 个方向，所以称之为 8 连通区域或八邻域。



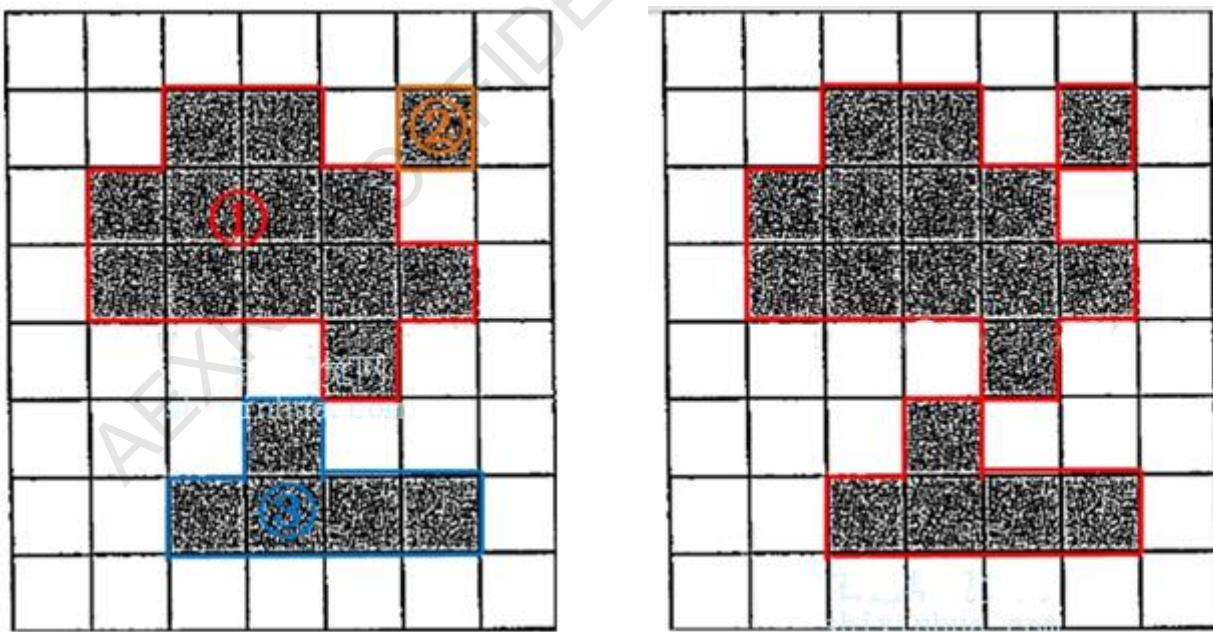
8 连通区域

八连通则定义为:

$$N8(p) = N4 \cup (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

对每一个值为 1 的点若其八连通有一个点的值也为 1,那么这两个点就归为一个物体。

从 8 连通的定义公式可以看到, 其为 N4 四连通并上右下、右上、左下、左上四点, 即四连通是八连通的子集。也就是说在图像处理中四连通的区域, 一定是八连通。



从左图可以看到, 4 连通意义上, 可以分成独立的三个区域, 8 连通则 1 个区域 (右图)

AXIVESREGION_T

【说明】

定义连通区域信息

【定义】

```
typedef struct axIVES_REGION_T {  
    AX_U32 u32Area; /* Represented by the pixel number */  
    AX_U16 u16Left; /* Circumscribed rectangle left border */  
    AX_U16 u16Top; /* Circumscribed rectangle top border */  
    AX_U16 u16Right; /* Circumscribed rectangle right border */  
    AX_U16 u16Bottom; /* Circumscribed rectangle bottom border */  
} AXIVESREGION_T;
```

【成员】

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

【注意】

无

【相关数据类型及接口】

无

AXIVES_CCBLOB_T

【说明】

定义连通区域标记的输出信息。

【定义】

```
typedef struct axIVES_CCBLOB_T {  
    AXIVES_CCL_MODE_E enMode;           /* [IN ]: CCL mode */  
    AX_U32 u32AreaThrs;                /* [IN ]: threshold of area pixels */  
    AX_U32 u32RegionNum;               /* [OUT]: Number of valid region */  
    AXIVES_REGION_T arrRegion[AXIVES_MAX_REGION_NUM];  
} AXIVES_CCBLOB_T;
```

【成员】

成员名称	描述
enMode	定义连通区域模式
u32AreaThrs	有效连通区域的面积阈值，面积小于这个阈值的不会被统计到 arrRegion 和 u32RegionNum
u32RegionNum	有效连通区域个数
arrRegion	连通区域信息

【注意】

无

【相关数据类型及接口】

无

SCD_CHN

【说明】

SCD 通道定义

【定义】

```
typedef AX_S32 SCD_CHN
```

【成员】

无

【注意】

无

【相关数据类型及接口】

无

AX_SCD_CHN_ATTR_T

【说明】

SCD 通道属性定义

【定义】

```
typedef struct axSCD_CHN_ATTR_T {
    SCD_CHN chn;
    AXIVES_RECT_T stArea;
    AX_U8 u8Thrd;           /* range: [1 - 100], recommend: 60 */
    AX_U8 u8Confidence;     /* range: [1 - 100], recommend: 60 */
} AX_SCD_CHN_ATTR_T;
```

【成员】

成员名称	描述
chn	SCD 通道
stArea	区域
u8Thrd	相关性阈值，范围[1 - 100]，数值越大表示越不相关，比如：100 标识两幅图像完全不相关。
u8Confidence	相关性阈值比例门限，范围[1 - 100]，表示区域内相关性小于 u8Thrd 阈值的 BLOCK 个数相对于区域内总 BLOCK 个数的百分比。

【注意】

无

【相关数据类型及接口】

无

AXIVES_BLK_SIZE_T**【说明】**

定义 BLOCK 大小

【定义】

```
typedef struct axIVES_BLK_SIZE_T {
    AX_U32 u32W;
    AX_U32 u32H;
} AXIVES_BLK_SIZE_T;
```

【成员】

成员名称	描述
u32W	宽度
u32H	高度

【注意】

无

【相关数据类型及接口】

无

4 错误码

错误码详见《55 - AX 软件错误码文档》文档。

5 日志

➤ 设置 IVES 日志等级

```
echo [id] [level] > /proc/ax_proc/logctl
```

id: [0-33]表示单模块 id, 如果设置所有模块则用 all 表示;

level: log 等级, 0-7 数字越小代表 Log 等级越高;

示例: 设置 IVES 模块 Log 等级为 4

```
echo 19 4 > /proc/ax_proc/logctl
```

➤ 设置日志目的地

Log 输出到文件:

```
echo target file > /proc/ax_proc/logctl
```

Log 输出到终端:

```
echo target console > /proc/ax_proc/logctl, Log 输出到终端, 不再保存到文件
```

关闭 Log:

```
echo target null > /proc/ax_proc/logctl
```